W3School Eclipse 教程

wizardforcel

Published with GitBook



目錄

W3School Eclipse 教程

来源：Eclipse 教程

整理：飞龙

Eclipse 教程



Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。

Eclipse 是 Java 的集成开发环境（IDE），当然 Eclipse 也可以作为其他开发语言的集成开发环境，如C，C++，PHP，和 Ruby 等。

Eclipse 附带了一个标准的插件集，包括Java开发工具（Java Development Kit，JDK）。

许可证

Eclipse 平台及其插件基于Eclipse公共许可证（EPL：Eclipse Public License）发布。

Eclipse公共许可证（简称EPL）是一种开源软件发布许可证，由Eclipse基金会应用于名下的集成开发环境Eclipse上。

发行版

从2006年起，Eclipse基金会每年都会安排同步发布。至今，同步发布主要在6月进行，并且会在接下来的9月及2月释放出SR1及SR2版本。

版本代号	平台版本	主要版本发行日期	**SR1**发行日期	**SR2**发行日期
Callisto	3.2	2006年6月26日	N/A	N/A
Europa	3.3	2007年6月27日	2007年9月28日	2008年2月29日
Ganymede	3.4	2008年6月25日	2008年9月24日	2009年2月25日
Galileo	3.5	2009年6月24日	2009年9月25日	2010年2月26日
Helios	3.6	2010年6月23日	2010年9月24日	2011年2月25日
Indigo	3.7	2011年6月22日	2011年9月23日	2012年2月24日
Juno	3.8及4.2	2012年6月27日	2012年9月28日	2013年3月1日
Kepler	4.3	2013年6月26日	2013年9月27日	2014年2月28日
Luna	4.4	2014年6月25日	2014年9月25日	N/A
Mars	4.5	2015年6月24日	N/A	N/A

Eclipse 安装

下载 Eclipse，下载地址为：http://www.eclipse.org/downloads/。 下载页面列出了不同语言的 Eclipse IDE，你可以根据自己需要下载。

Eclipse 的每个安装包都不同， Java 开发人员通常使用 Eclipse IDE for Java Developers 来开发 Java 应用。

列表右侧提供了Windows，Linux 和 Mac 操作系统及对应的32位与64位的安装包，你可以根据自己的系统情况选择合适的包下载。

Eclipse Luna (4.4.1) Release for Windows

Eclipse IDE for Java Developers, 154 MB
Downloaded 2,581,552 Times
The essential tools for any Java developer, including a Java IDE, a CVS client, Git client, XML Editor, Mylyn, Maven integration...
Windows 32 Bit
Windows 64 Bit

Package Solutions
Filter Packages

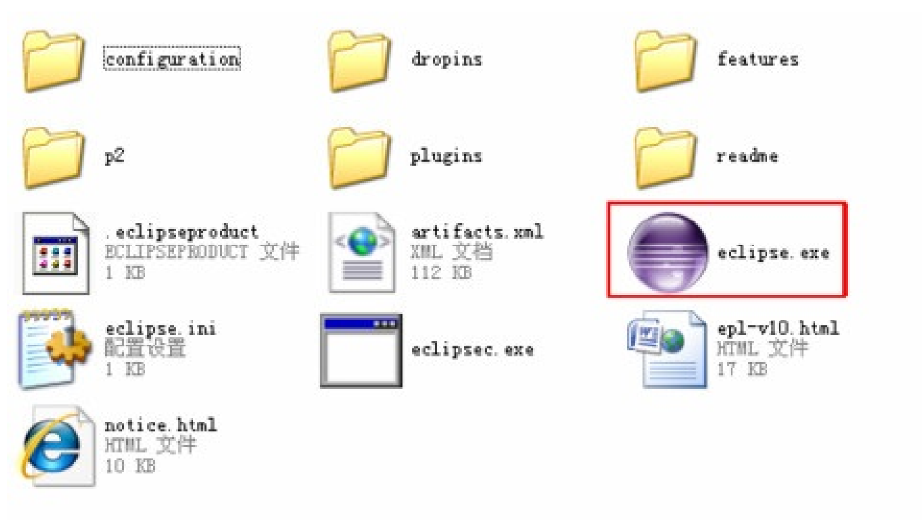
Eclipse IDE for Java EE Developers, 254 MB
Downloaded 1,570,255 Times
Tools for Java developers creating Java EE and Web applications, including a Java IDE, tools for Java EE, JPA, JSF, Mylyn...
Windows 32 Bit
Windows 64 Bit

JRebel for Eclipse IDE
See Java Code Changes Instantly. Save Time. Reduce Stress. Finish Projects Faster!
Promoted Download

安装 Eclipse

Eclipse 是基于 Java 的可扩展开发平台，所以安装 Eclipse 前你需要确保你的电脑已安装 JDK，JDK 安装可以查看我们的Java开发环境配置。

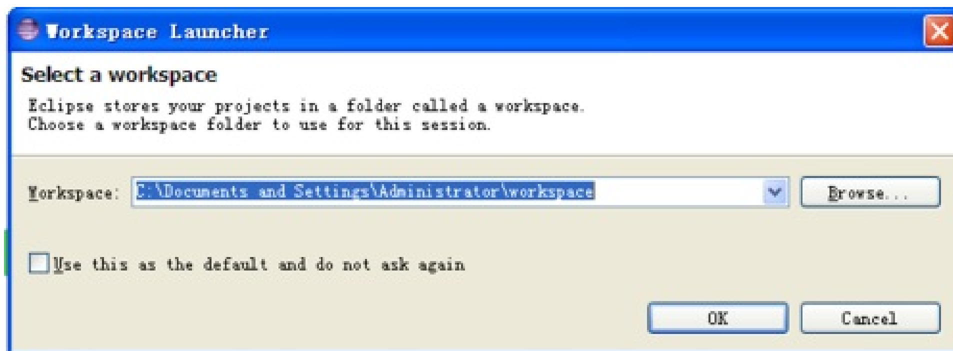
Eclipse 的安装非常简单，你只需要下载压缩包，解压完毕后即可使用，进入文件夹，红框如图所示就是eclipse的启动程序



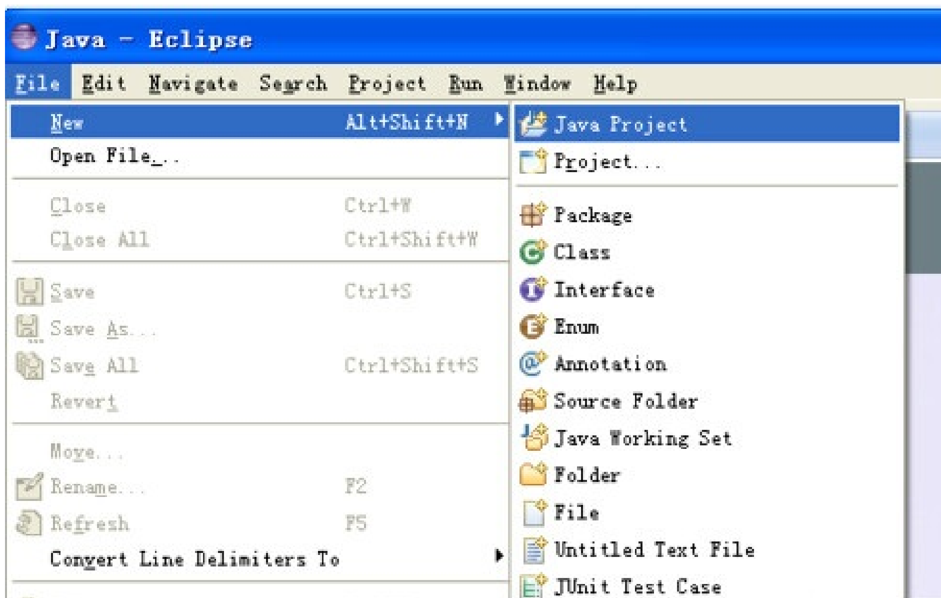
若你打开Eclipse的时候发现如下的对话框，则说明你的电脑未安装 JDK 环境。



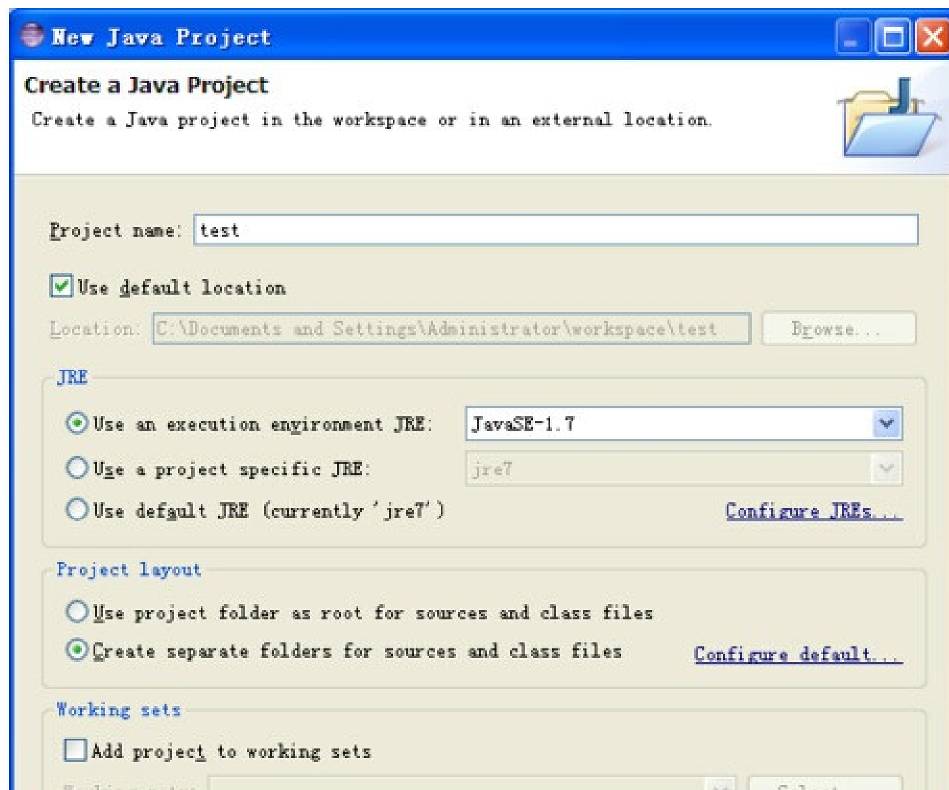
第一次打开需要设置工作环境，你可以指定工作目录，或者使用默认的C盘工作目录，点击ok 按钮。



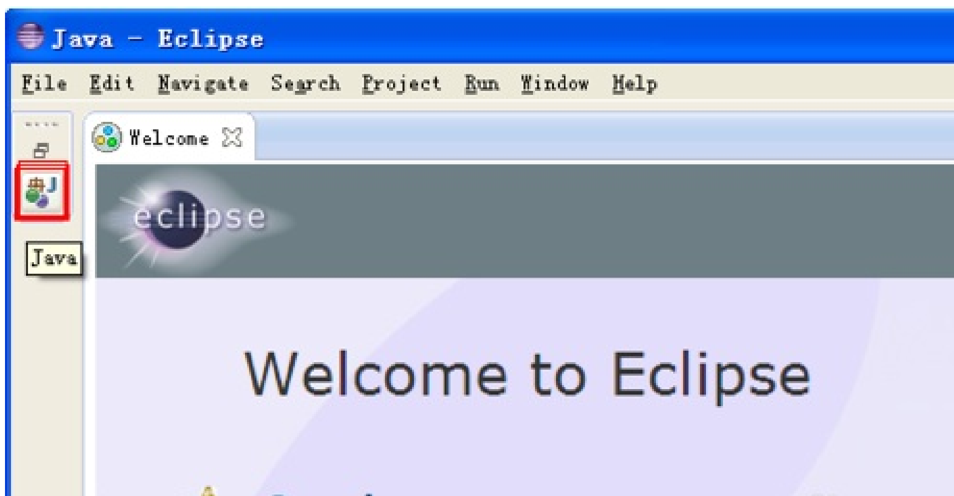
创建一个项目：选择file--New--java Project，如图:



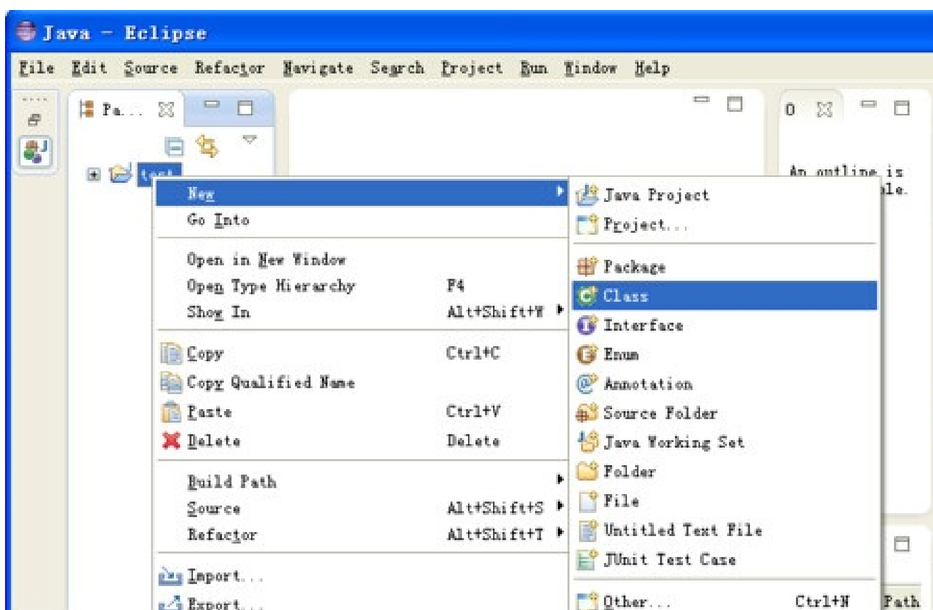
输入项目名称，比如我输入test，然后点击finish



完成项目的创建，点击红框里的小图标，如图：



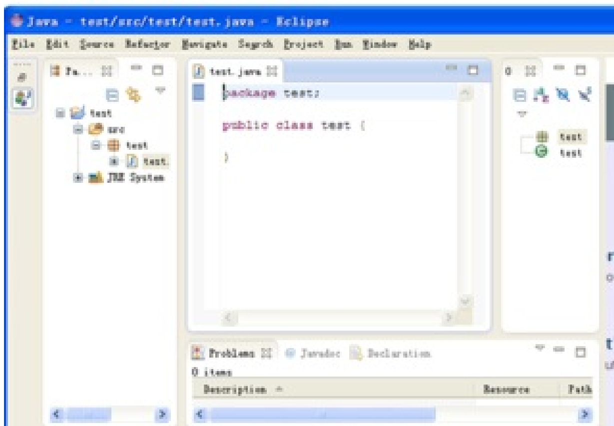
然后在左侧菜单选择test项目，右键--new--class



键入类名，如输入 test，如图，然后点击finish



这样在代码框里面你就可以开始输入代码啦！

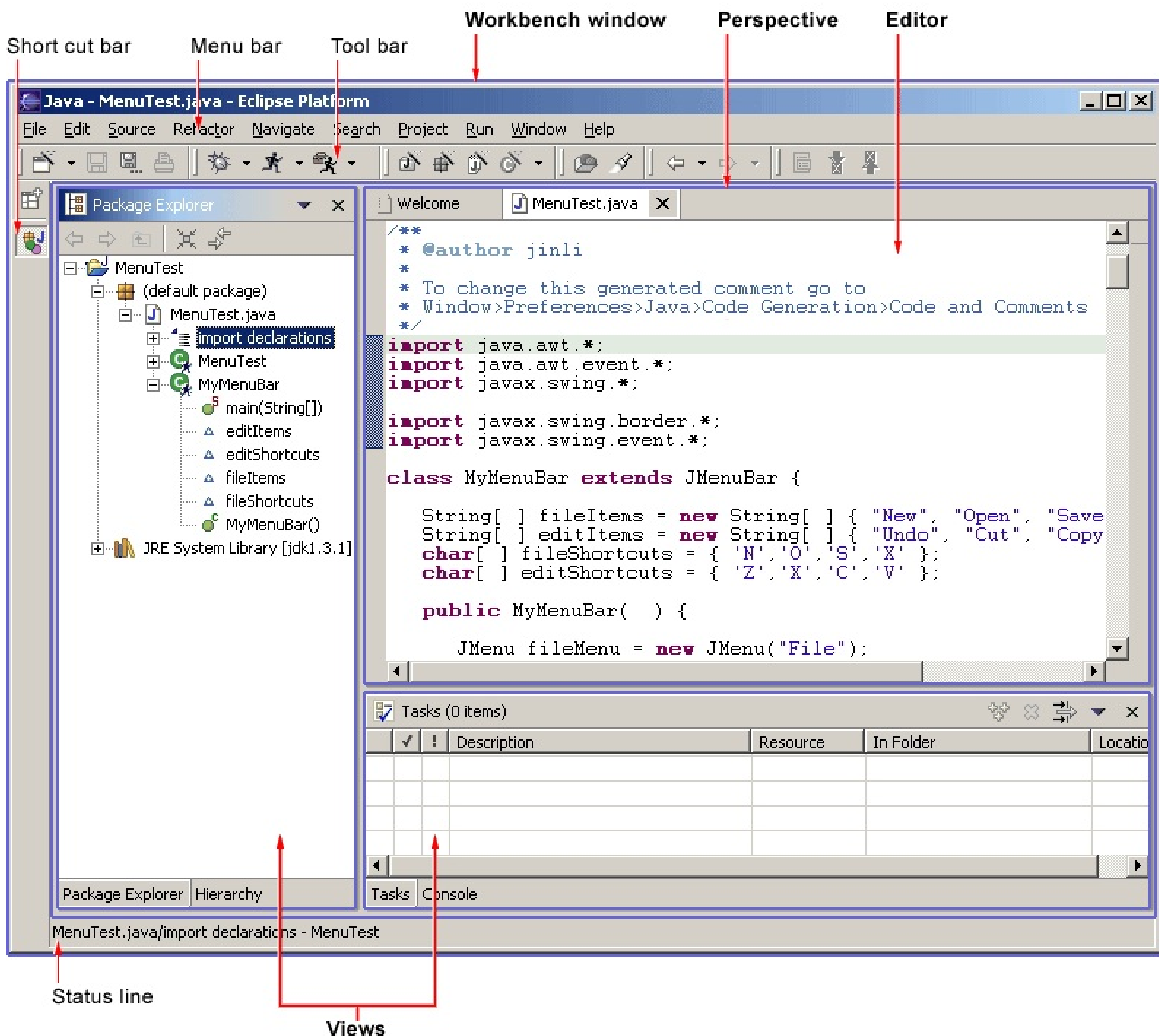


Eclipse 窗口说明

Eclipse 工作台(Workbench)

首先，让我们来看一下Eclipse 作台用户界面，和它里面的各种组件。

工作台是多个窗口的集合。每个窗口包含菜单栏，工具栏，快捷方式栏，以及一个或者多个透视图。



透视图是一个包含一系列视图和内容编辑器的可视容器。

视图完全存在于某个透视图中而且不能被共享，而任何打开的内容编辑器可以在透视图间共享。

如果两个或者多个透视图打开了同样的视图，他们共享这个视图的同一个实例，虽然在不同透视图之间视图的布局可能不同。

对于不同的工作台窗口中的透视图，编辑器和视图都不能共享。

一个透视图就好像是一本书里面的一页。它存在在一个窗口中，并且和其他透视图一起存在，和书中的一页一样，每次你只能看到一个透视图。

工作台的主菜单栏通常包括File，Edit，Navigate，Project，Window，Help这些顶层菜单。

其他的顶层菜单位于Edit和Project菜单之间，往往是和上下文相关，这个上下文包括当前活动的透视图，最前面的编辑器以及活动视图。

在File菜单中，你可以找到一个New子菜单，它包括Project，Folder，File的创建菜单项。

File 菜单也包含Import and Export菜单项，用来导入文件到Wrokbench中，以及导出它们。

在Edit菜单中，你可以找到象Cut，Copy，Paste，和Delete这些命令。这些命令称为全局命令，作用于活动部件。

也就是说，如果当Navigator活动时使用Delete命令, 实际操作是由Navigator完成的。

在Project菜单中，你可以找到和项目相关的命令，比如Open Project，Close Project和Rebuild Porject等。

在Run菜单中，你可以看到和运行，调试应用代码相关的命令，以及启动象Ant脚本这样的外部工具。

在Window菜单中，你可以找到Open Perspective子菜单，根据你开发任务的需要打开不同的透视图。

你也能看到透视图 布局管理菜单栏。Show View子菜单用来在当前的Workbench窗口中增加视图。

另外，你可以通过首选项菜单项来修改工作台的功能首选项配置。

如果你是插件开发者，你可以为平台提供新的视图，编辑器，向导，菜单和工具项。 这些东西都是用XML来定义的，插件一旦注册后，就可以和平台中已经存在的组件无缝地集成在一起。

Eclipse 多窗口

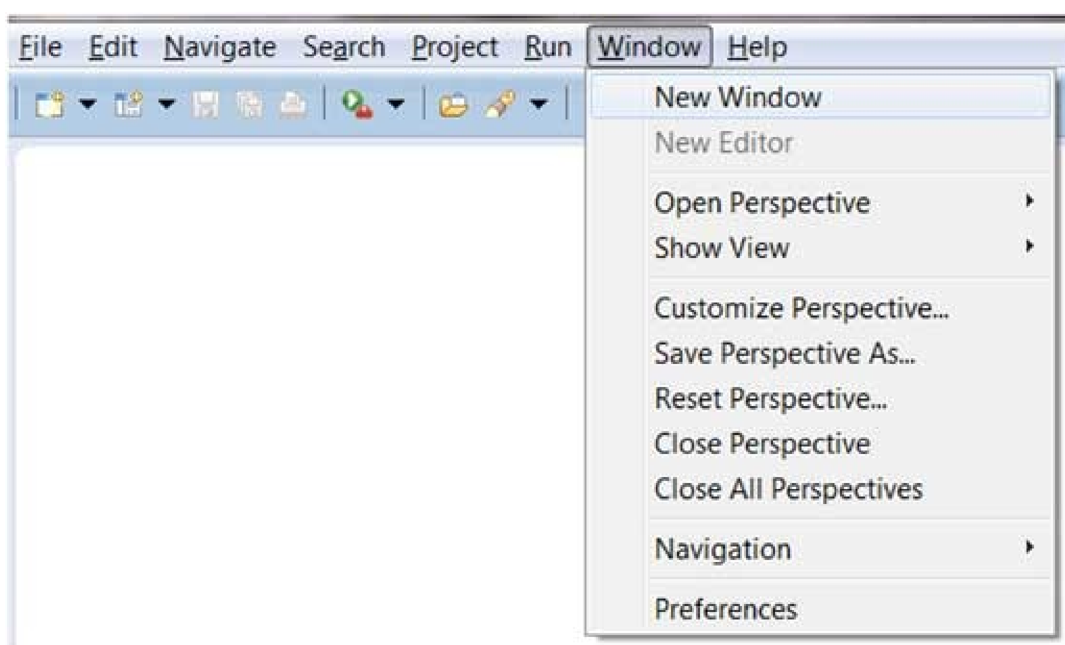
Eclipse 可以同时开启多个窗口，在 菜单栏选择： Window -> New Window 来开启多窗口。

多个窗口的切换你可以使用 Alt + Tab 来回切。

Eclipse 菜单

Eclipse 查看的菜单栏通常包含以下几个菜单：

- File 菜单
- Edit 菜单
- Navigate 菜单
- Search 菜单
- Project 菜单
- Run 菜单
- Window 菜单
- Help 菜单



通过 Eclipse 插件你可以添加新的菜单和菜单项。

菜单描述

菜单名	描述
File	File 菜单运行你打开文件，关闭编辑器，保存编辑的内容，重命名文件。 此外还可以导入和导出工作区的内容及关闭 Eclipse。
Edit	Edit 菜单有复制和粘贴等功能。
Source	只有在打开 java 编辑器时 Source 菜单才可见。 Source 菜单关联了一些关于编辑 java 源码的操作。
Navigate	Navigate 菜单包含了一些快速定位到资源的操作。
Search	Search 菜单可以设置在指定工作区对指定字符的搜索。
Project	Project 菜单关联了一些创建项目的操作。
Run	Run 菜单包含了一些代码执行模式与调试模式的操作。
Window	Window 菜单允许你同时打开多个窗口及关闭视图。 Eclipse 的参数设置也在该菜单下。
Help	Help 菜单用于显示帮助窗口，包含了 Eclipse 描述信息，你也可以在该菜单下安装插件。

Eclipse 也可以自定义菜单，自定义菜单的详细介绍可以查看 Eclipse 透视图。

Eclipse 视图

关于视图

Eclipse视图允许用户以图表形式更直观的查看项目的元数据。 例如，项目导航视图中显示的文件夹和文件图形表示在另外一个编辑窗口中相关的项目和属性视图。

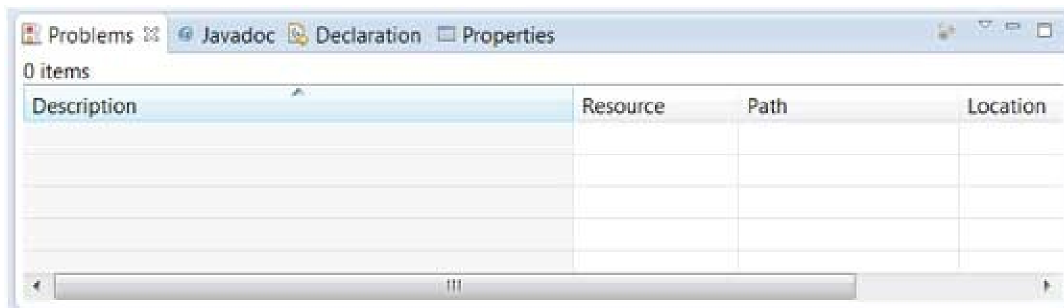
Eclipse 透视图(perspective) 可以显示任何的视图和编辑窗口。

所有的编辑器实例出现在一个编辑器区域内，可以通过文件夹视图查看。

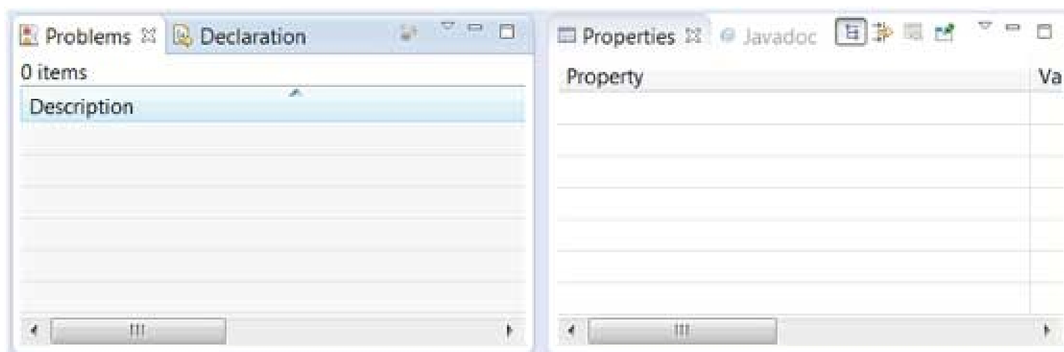
一个工作台窗口可以显示任意数量的文件夹视图。每个文件夹视图可以显示一个或多个视图。

组织视图

下图显示了文件夹视图的四个视图。

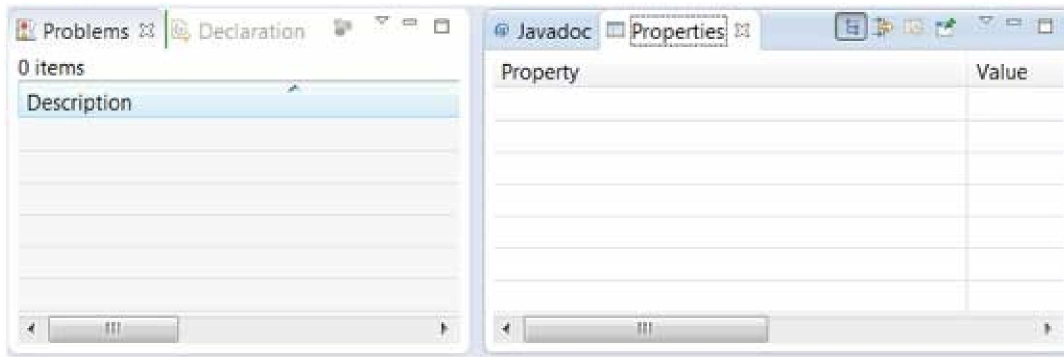


下图在两个文件夹视图中显示四个视图。



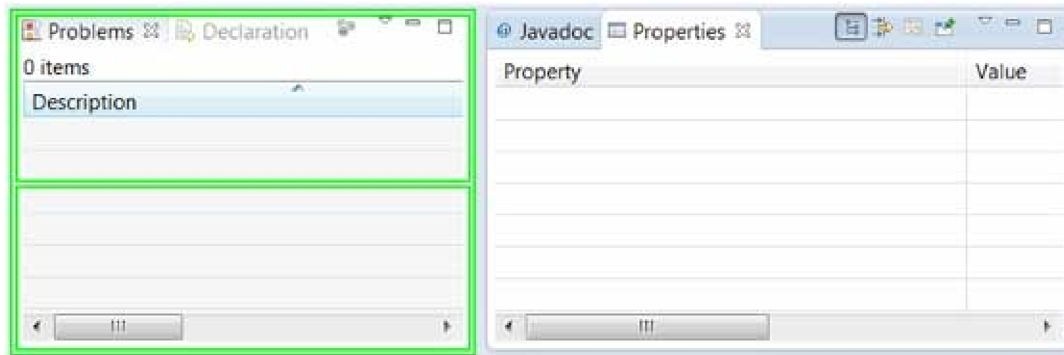
移除视图 views

视图从一个文件夹视图移动到另外一个文件夹视图只需要点击视图标题并推动视图工具区域到另外一个文件夹视图。



创建文件夹视图

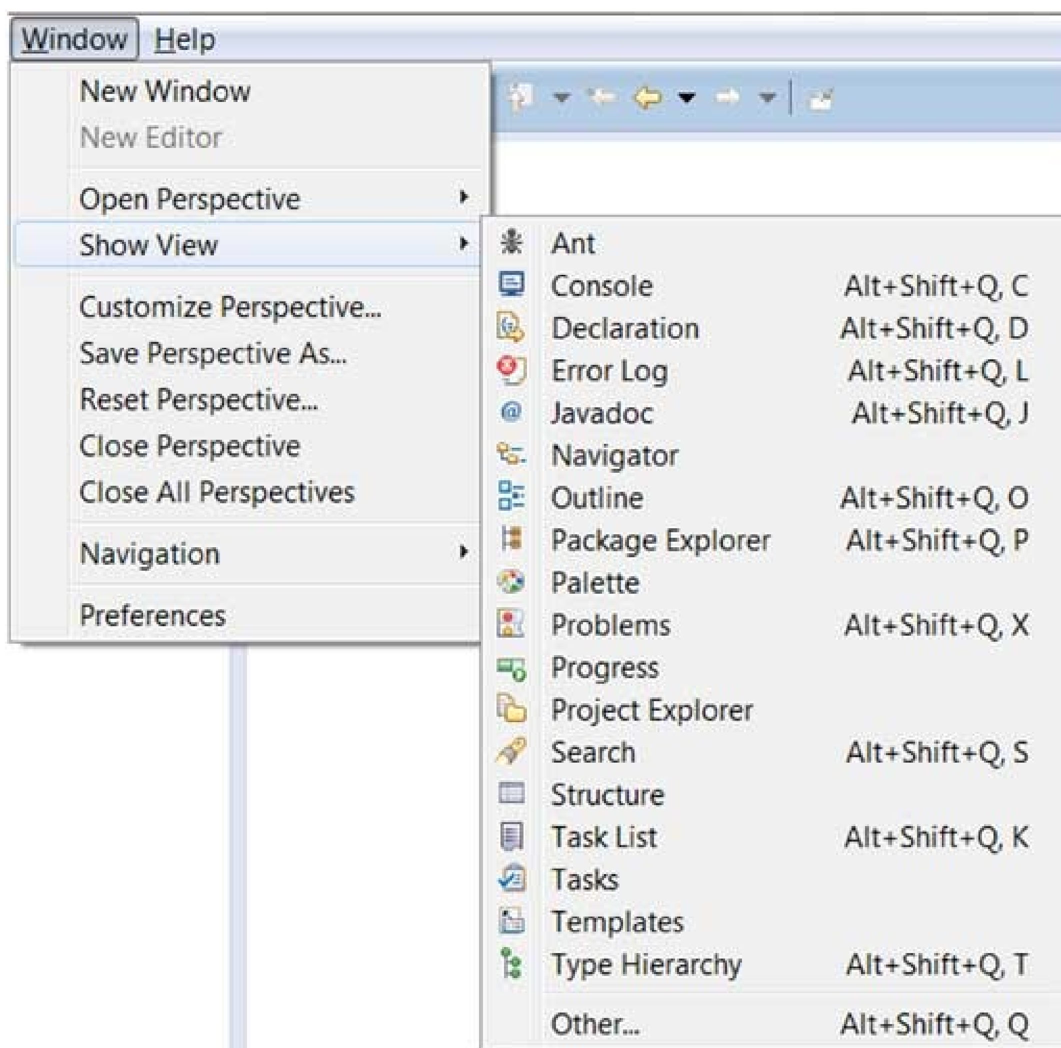
文件夹视图可以通过移动视图标题栏到编辑去外或移动标题栏到另外一个文件夹视图来动态创建。 下图中如果你拖动了绿色线框内的标题栏意味着一个新的文件夹视图将被创建。



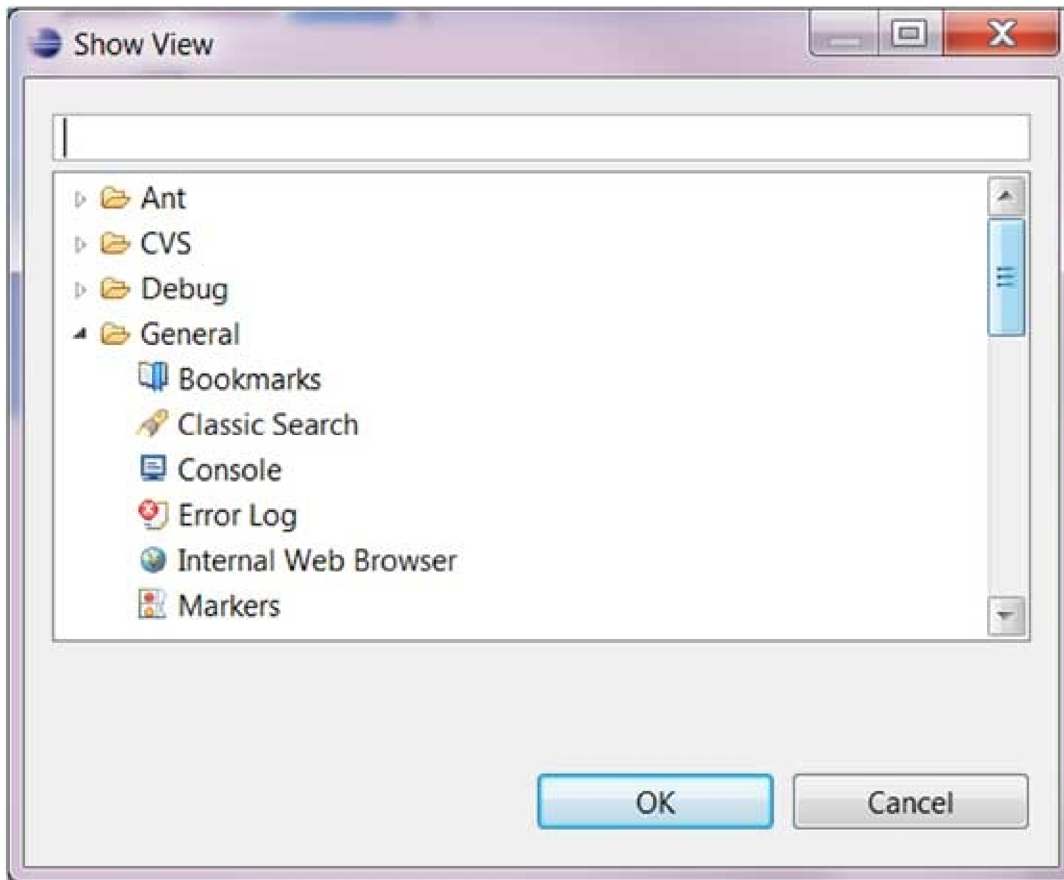
移动拖动图标到窗口的底部，您可以创建一个横跨窗口整个宽度的视图文件夹。移动拖动图标到窗口的左边或右边，您可以创建一个横跨窗口的整个高度视图文件夹。

操作视图

你可以在 Window 菜单中点击 "Show View" 选项打开其他视图。



点击 "Other" 菜单选项会弹出一个 "Show View" 对话框，对话框中你可以查找和激活视图。



视图通过各个分类来组织。你可以通过搜索框快速查找视图。 然后打开视图并选择，点击 "OK" 按钮即可。

什么是透视图？

透视图是一个包含一系列视图和内容编辑器的可视容器。默认的透视图叫 java。

Eclipse 窗口可以打开多个透视图，但在同一时间只能有一个透视图处于激活状态。

用户可以在两个透视图之间切换。

操作透视图

通过"Window"菜单并选择"Open Perspective > Other"来打开透视图对话框。



透视图对话框中显示了可用的透视图列表。

该透视图列表也可以通过工具栏上的透视图按钮来打开 ( )。

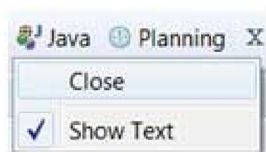
视图切换

大都数情况下 java 开发者会使用 Java 透视图和 Debug 透视图。你可以通过工具条上的透视图名称来自由切换。



关闭透视图

工具条上右击透视图名及选择"Close"项即可关闭透视图。



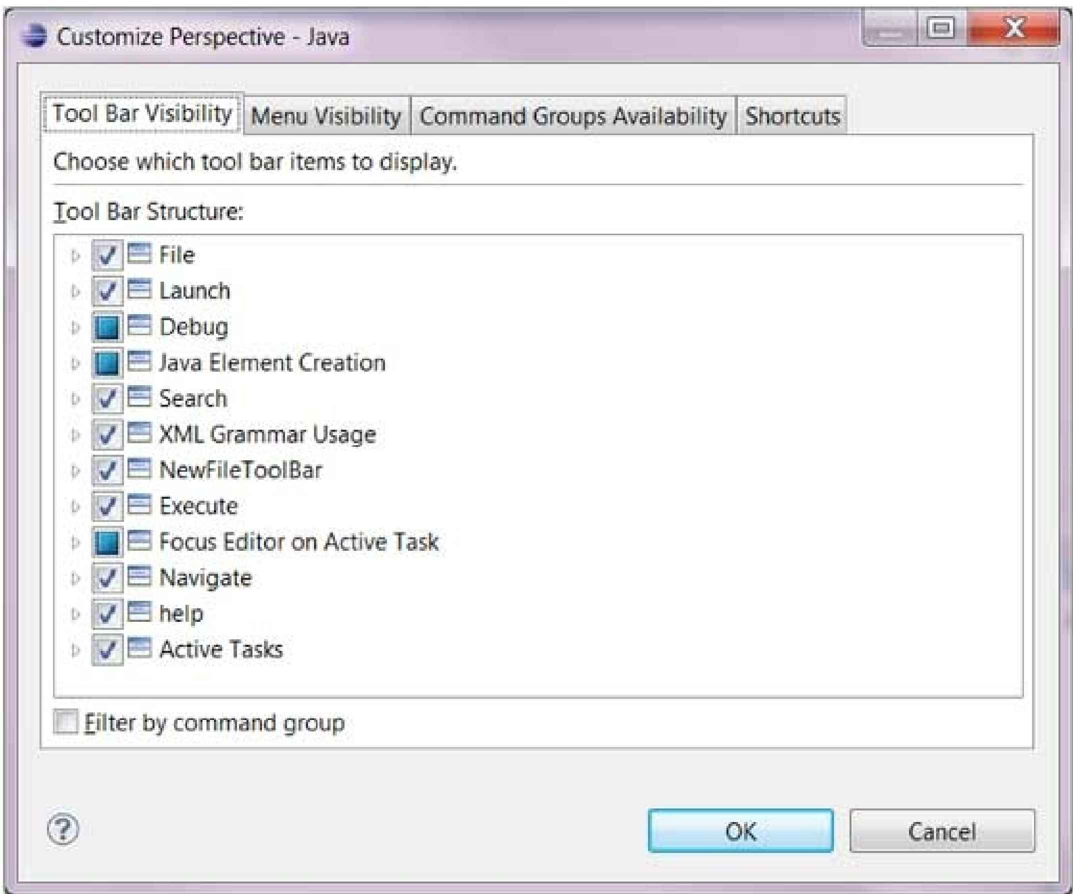
自定义透视图

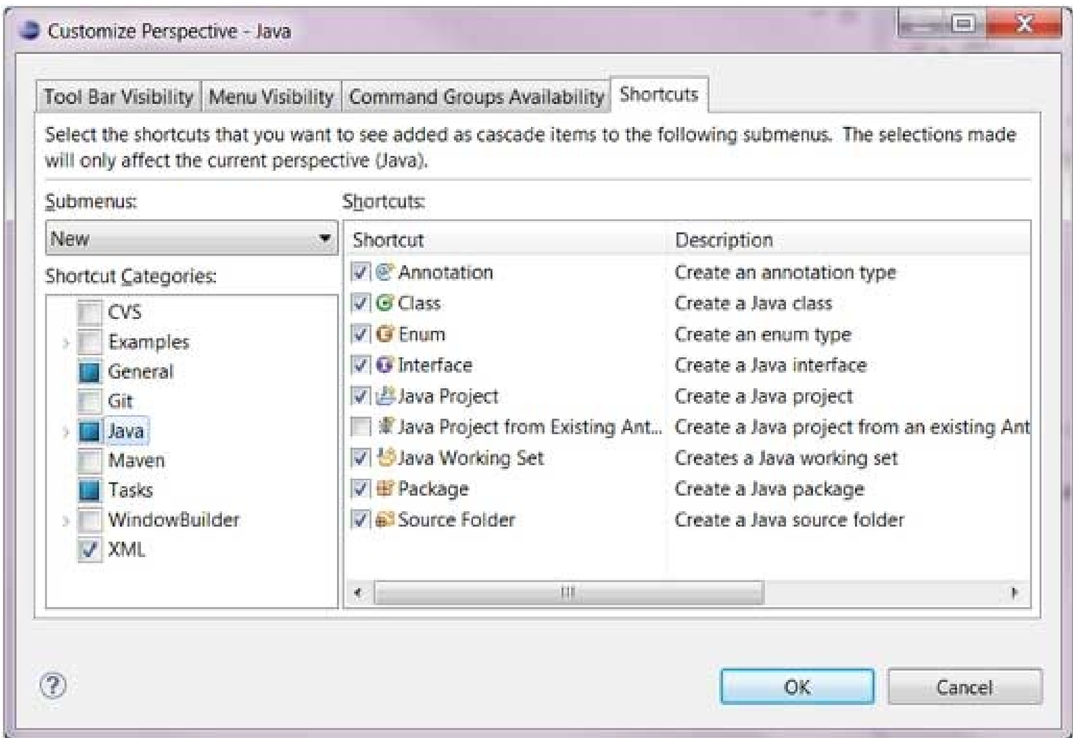
我们可以通过自定义透视图窗口来设置我们想要的透视图。

- 点击菜单栏上的 "Windows" => "Customize Perspective" => 弹出窗口，可以

在"Submenus"里面选择你要设置的内容。

- "New" => 设置你的新建菜单，可以把你平时需要新建的最常用的文件类型选中。
- "Show Views" 在你自定义的这个视图的布局，也就是切换到你自己的视图后，会出现哪些窗口。根据自己习惯进行设置。
- Open Perspective" => 切换视图菜单中，出现哪些可以选择的视图。
- 都设置好以后，保存你的自定义透视图。Windows => Save Perspective as，然后为你自定义的透视图取一个名字，保存。


Customize Perspective - Java
Tool Bar Visibility
Menu Visibility
Command Groups Availability
Shortcuts
Choose which tool bar items to display.
Tool Bar Structure:
File
Launch
Debug
Java Element Creation
Search
XML Grammar Usage
NewFileToolBar
Execute
Focus Editor on Active Task
Navigate
help
Active Tasks
Filter by command group
OK
Cancel

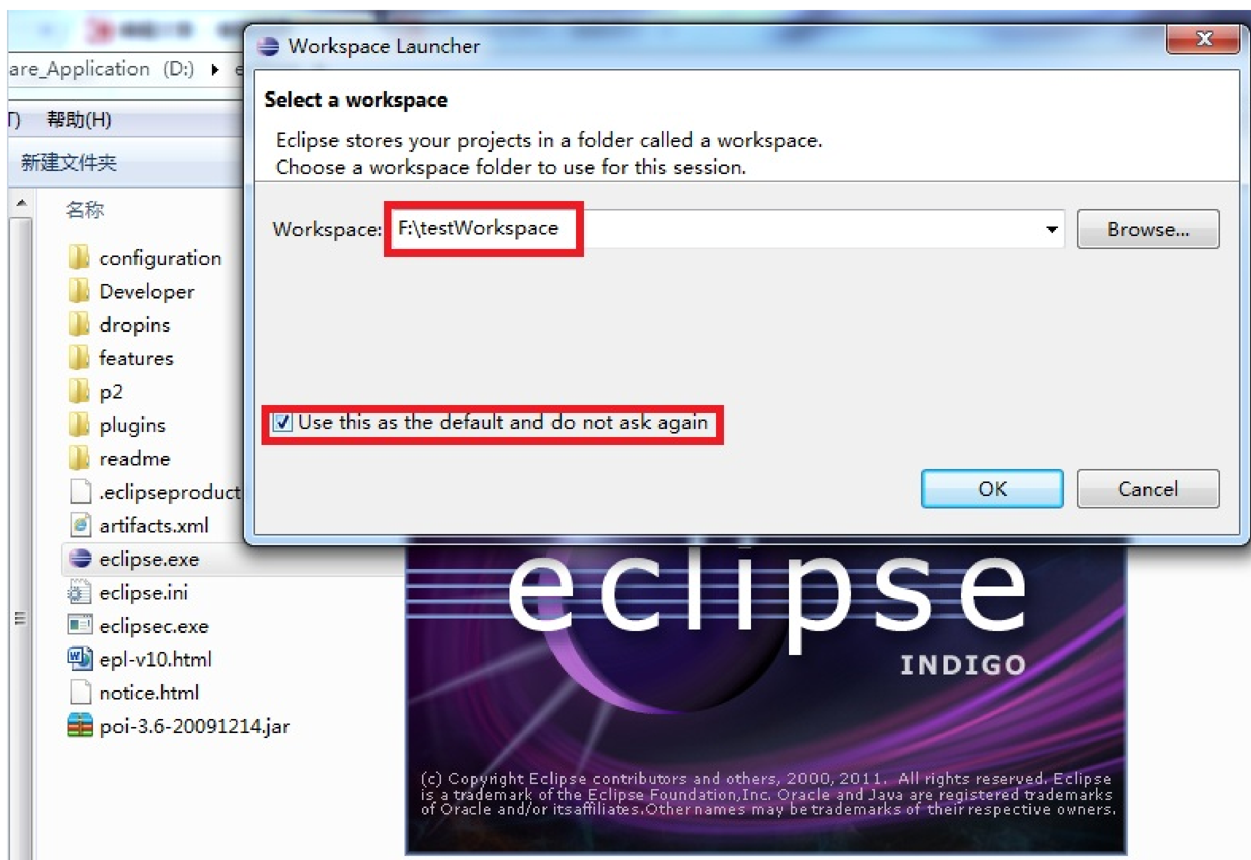

Customize Perspective - Java
Tool Bar Visibility
Menu Visibility
Command Groups Availability
Shortcuts
Select the shortcuts that you want to see added as cascade items to the following submenus. The selections made will only affect the current perspective (Java).
Submenus:
New
Shortcut Categories:
CVS
Examples
General
Git
Java
Maven
Tasks
WindowBuilder
XML
Shortcuts:
Shortcut
Description
Annotation
Create an annotation type
Class
Create a Java class
Enum
Create an enum type
Interface
Create a Java interface
Java Project
Create a Java project
Java Project from Existing Ant...
Create a Java project from an existing Ant
Java Working Set
Creates a Java working set
Package
Create a Java package
Source Folder
Create a Java source folder
OK
Cancel

Eclipse 工作空间(Workspace)

eclipse 工作空间包含以下资源：

- 项目
- 文件
- 文件夹

项目启动时一般可以设置工作空间，你可以将其设置为默认工作空间，下次启动后无需再配置：

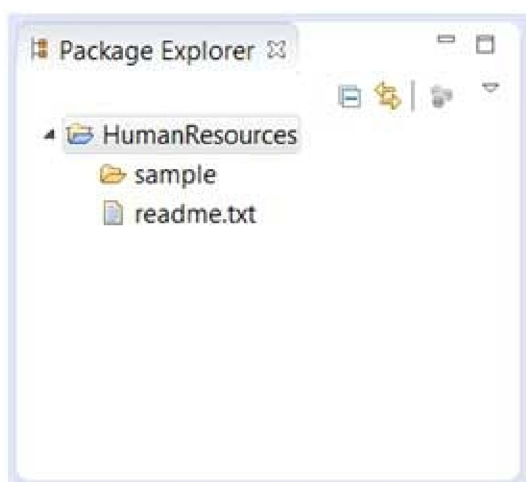


工作空间(Workspace)有明显的层次结构。 项目在最顶级，项目里头可以有文件和文件夹。

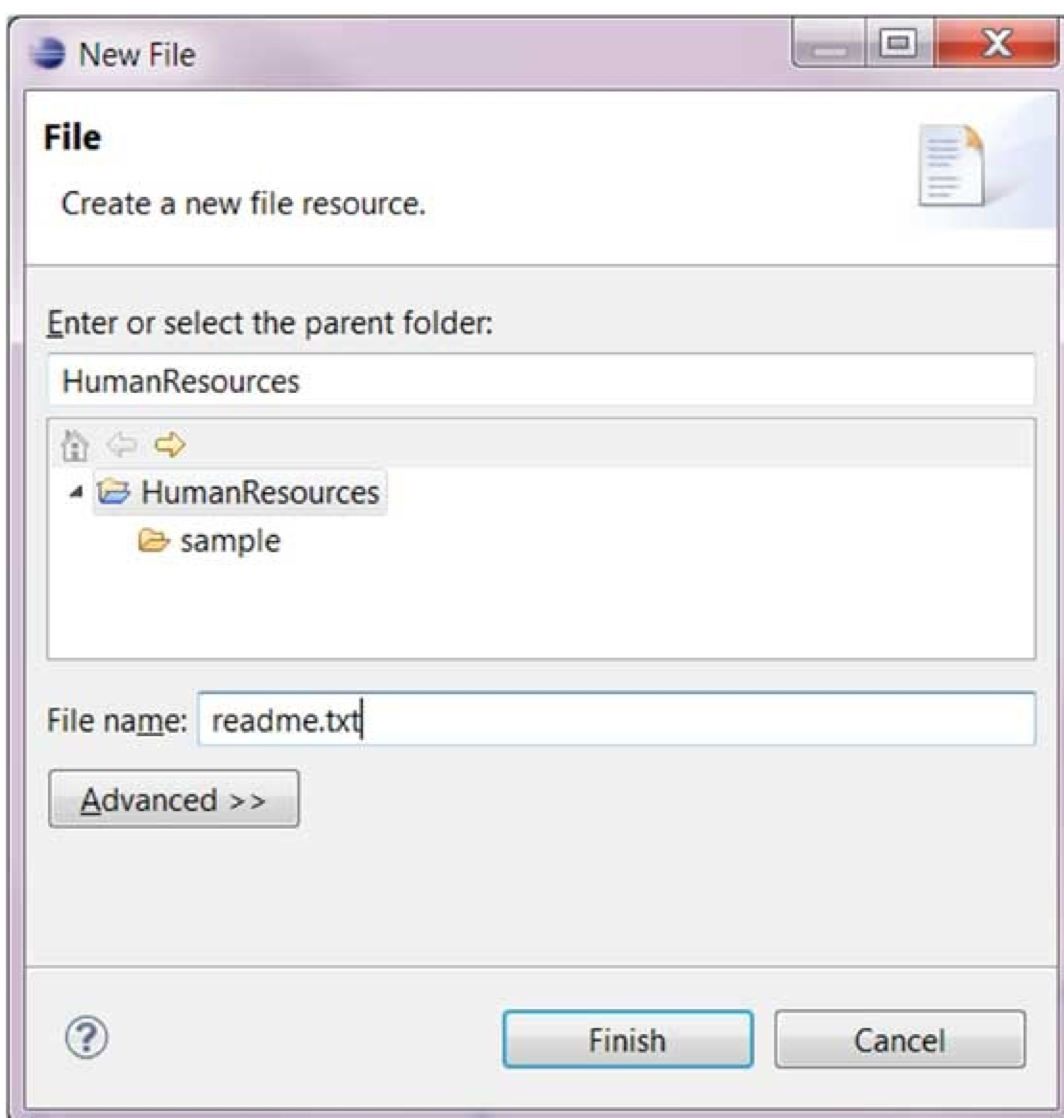
插件可以通过资源插件提供的API来管理工作空间的资源。

管理工作空间(Workspace)

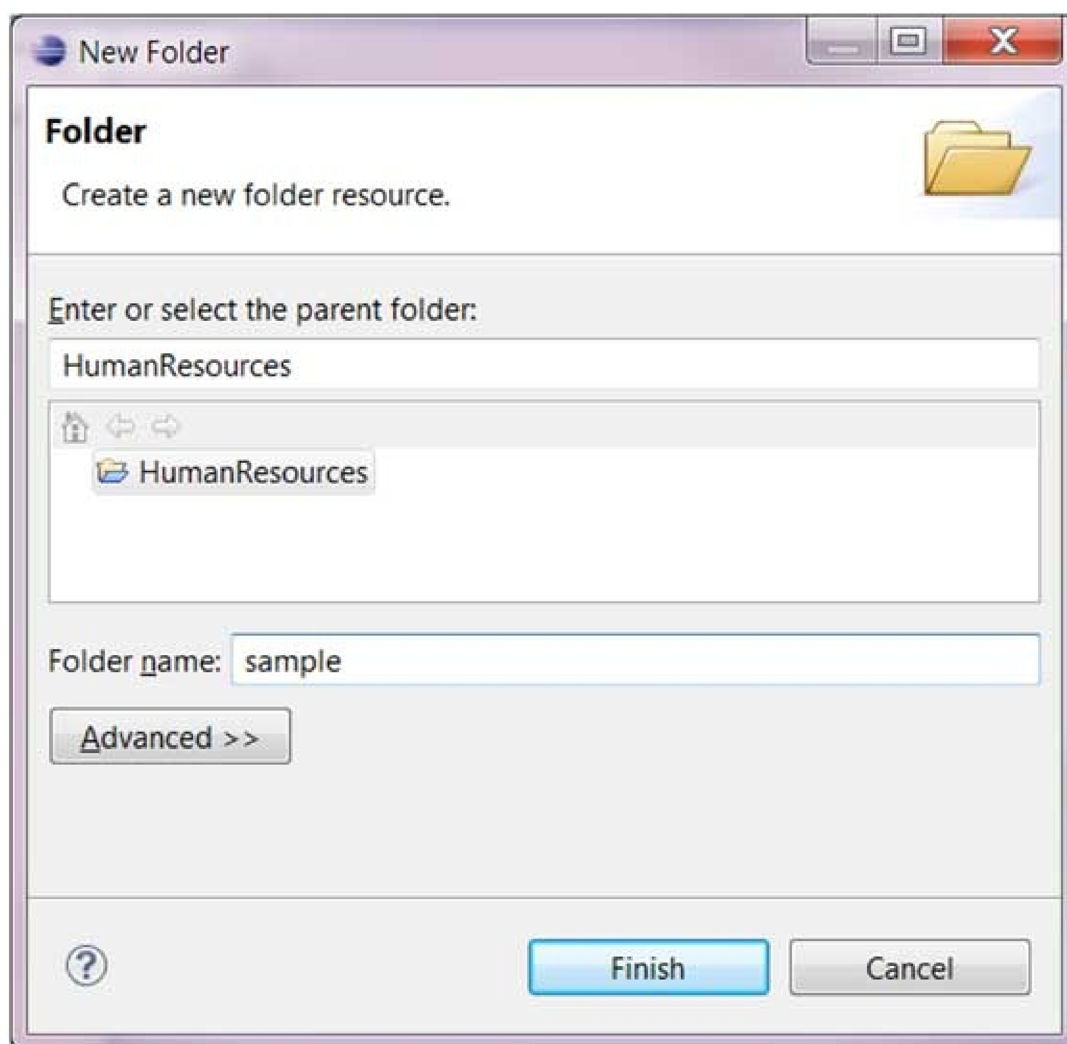
用户通过使用视图，编辑器和向导功能来创建和管理工作空间中的资源。其中，显示工作区的内容很多意见中的Project Explorer视图。显示项目工作空间内容的视图是Project Explorer视图。



文件创建向导(File > New > File)。

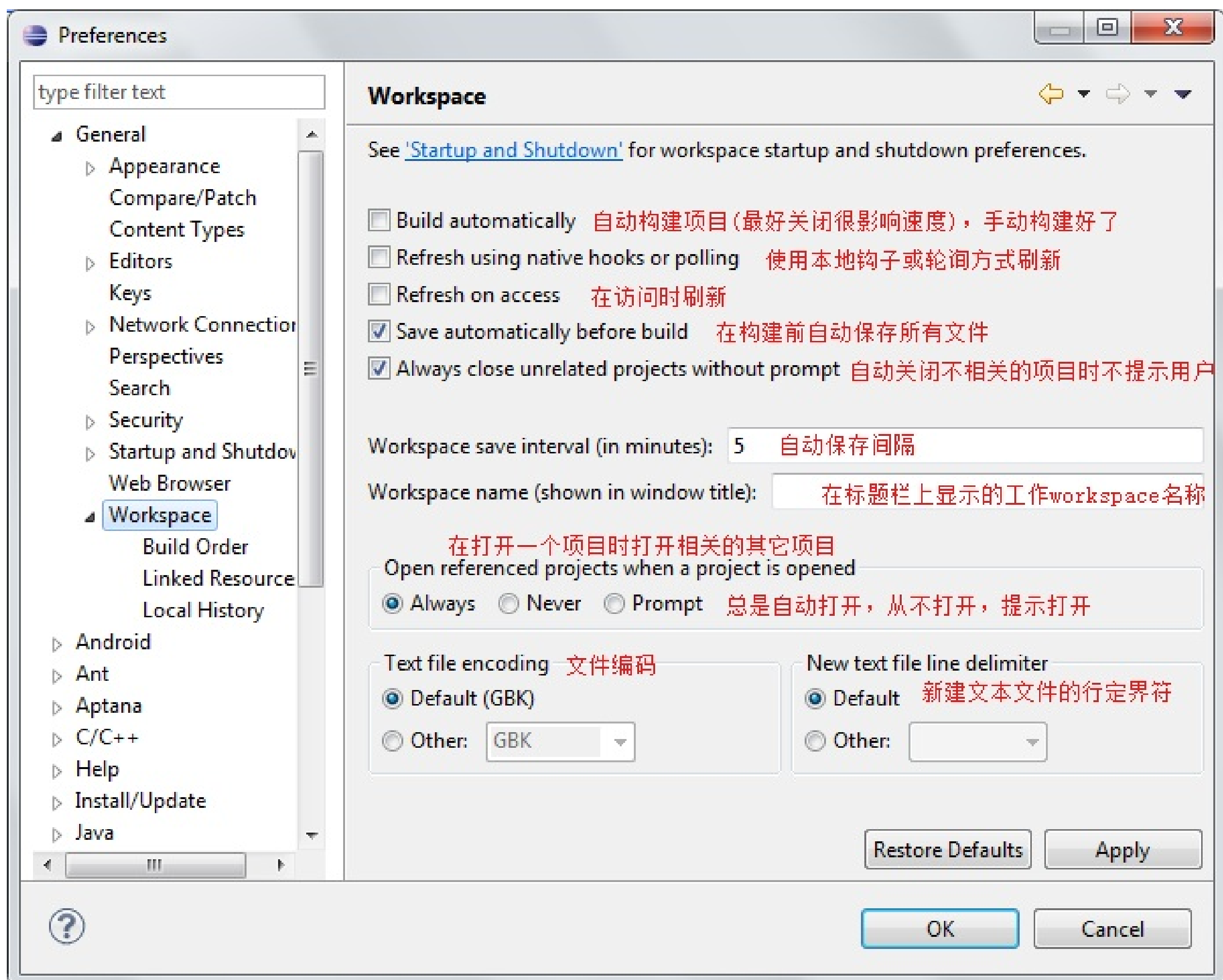


文件夹(Folder)创建向导(File > New > Folder)。



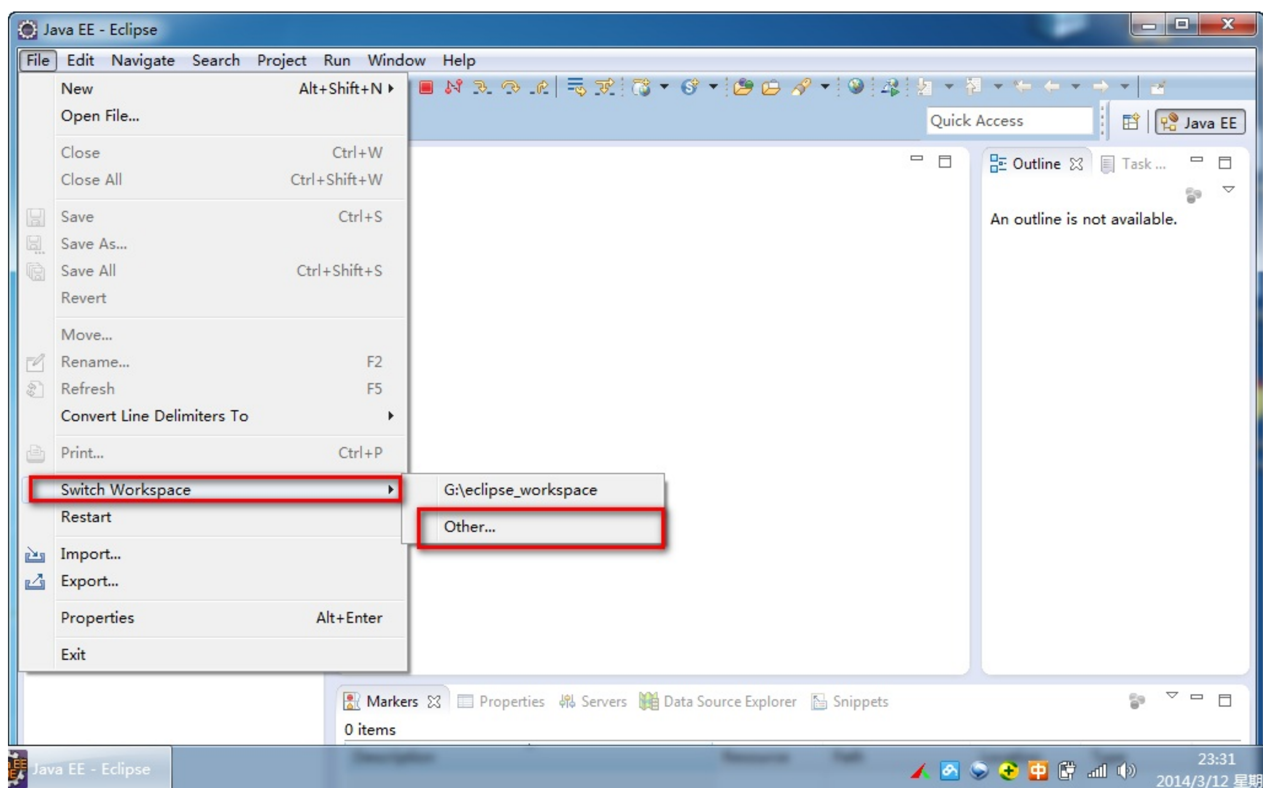
工作空间（workspace）设置

在菜单栏上选择 "Window" => "preferences..." => "General"=>"Workspace"，设置说明如下图：



Eclipse切换工作空间(workspace)

Eclipse切换工作空间可以选择菜单栏中选择 "File" => "switch workspace"：


Java EE - Eclipse
File Edit Navigate Search Project Run Window Help
New Alt+Shift+N
Open File...
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... Ctrl+P
Switch Workspace
Restart
Import...
Export...
Properties Alt+Enter
Exit
G:\eclipse_workspace
Other...
Quick Access
Java EE
Outline
Task ...
An outline is not available.
Markers
Properties
Servers
Data Source Explorer
Snippets
0 items
Java EE - Eclipse
23:31
2014/3/12

Eclipse 创建 Java 项目

打开新建 Java 项目向导

通过新建 Java 项目向导可以很容易的创建 Java 项目。打开向导的途径有：

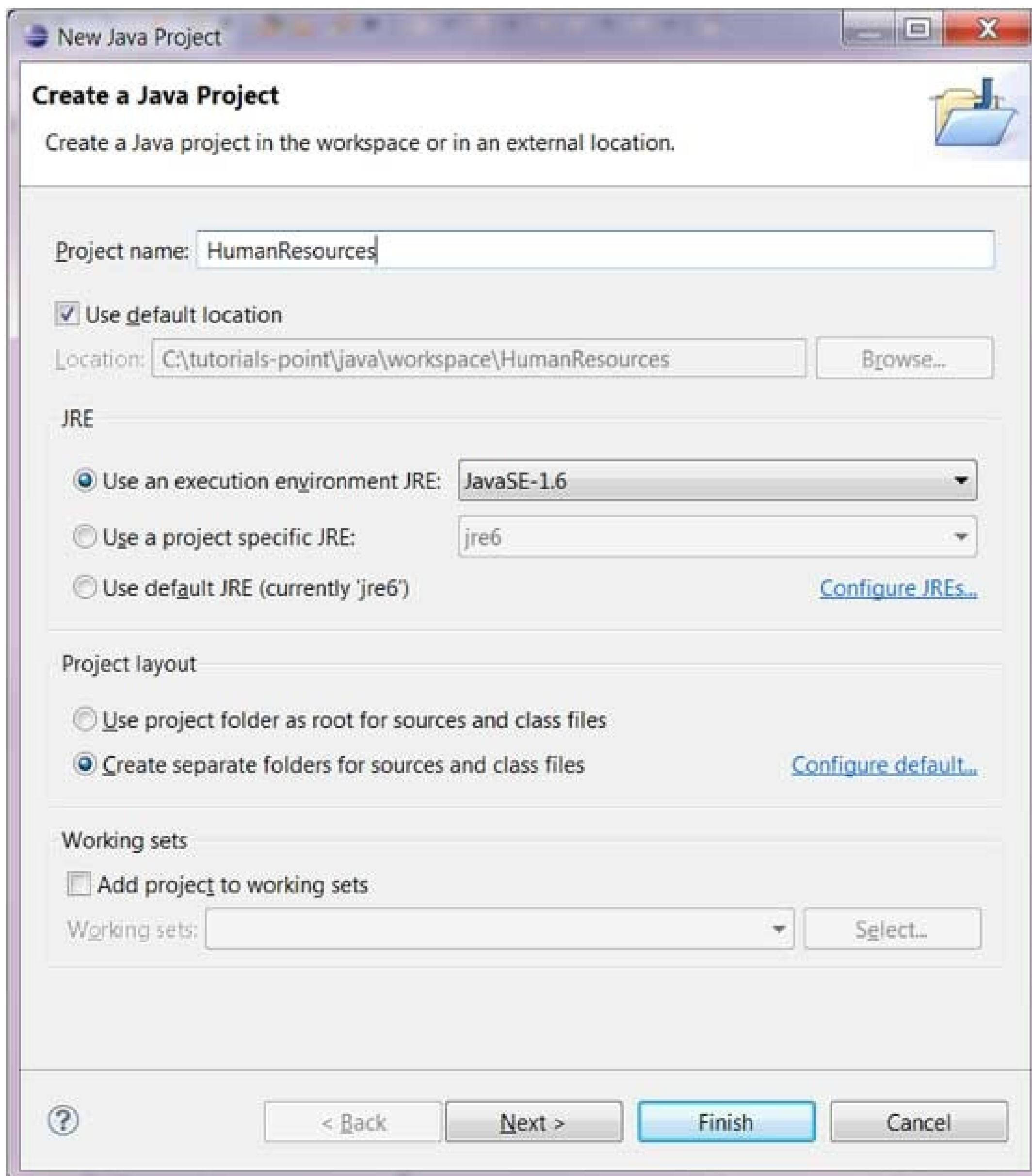
- 通过点击 "File" 菜单然后选择 New > Java Project
- 在项目浏览器(Project Explorer)窗口中鼠标右击任一地方选择 New > Java Project
- 在工具条上点击新建按钮 ( ) 并选择 Java Project

使用新建 Java 项目向导

新建 Java 项目向导有两个页面。

第一个页面:

- 输入项目名称（Project Name 栏中）
- 选择 Java Runtime Environment (JRE) 或直接采用默认的
- 选择项目布局（Project Layout），项目布局决定了源代码和 class 文件是否放置在独立的文件夹中。 推荐的选项是为源代码和 class 文件创建独立的文件夹。

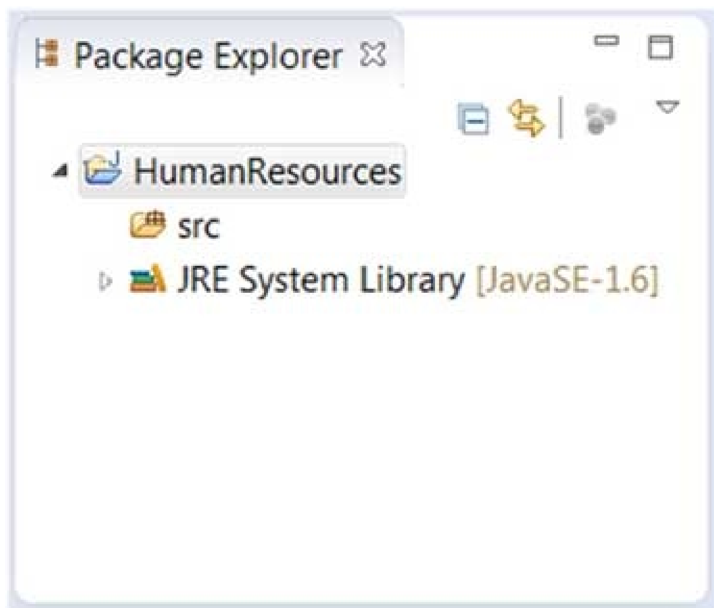


你可以点击"Finish" 按钮来创建项目或点击"Next" 按钮来修改 java 构建的配置。

第二个页面 Java 构建路径设置（Java Build Settings ），该页面我们可以配置项目的依赖关系及额外的 jar 包。

查看新建项目

Package Explorer 显示了新建的 Java 项目。项目图标中的 "J" 字母表示 Java 项目。 文件夹图标表示这是一个 java 资源文件夹。


Package Explorer
HumanResources
src
JRE System Library [JavaSE-1.6]

Eclipse 创建 Java 包

打开新建 Java 包向导

你可以使用新建 Java 包向导来创建 Java 包。Java 包向导打开方式有：

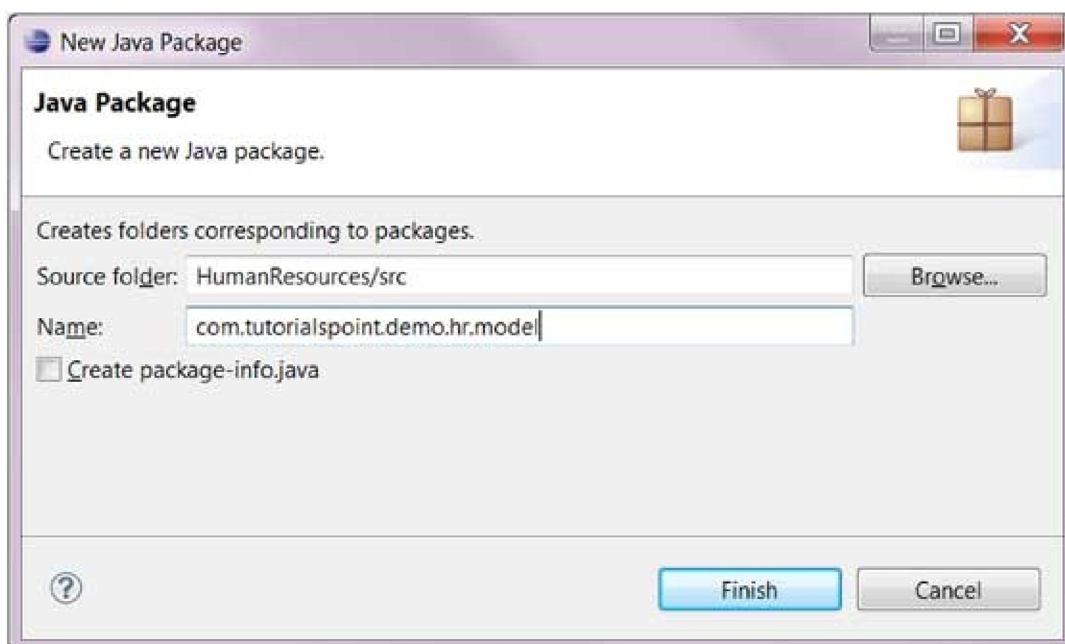
- 通过点击 "File" 菜单并选择 New > Package
- 在 Package Explorer 中通过右击鼠标选择 > Package
- 在工具条上点击包按钮( )

如果你要创建子包，在打开创建 Java 包向导前选择好父包，这样在名称字段就有了父包的值。

使用创建 Java 包向导

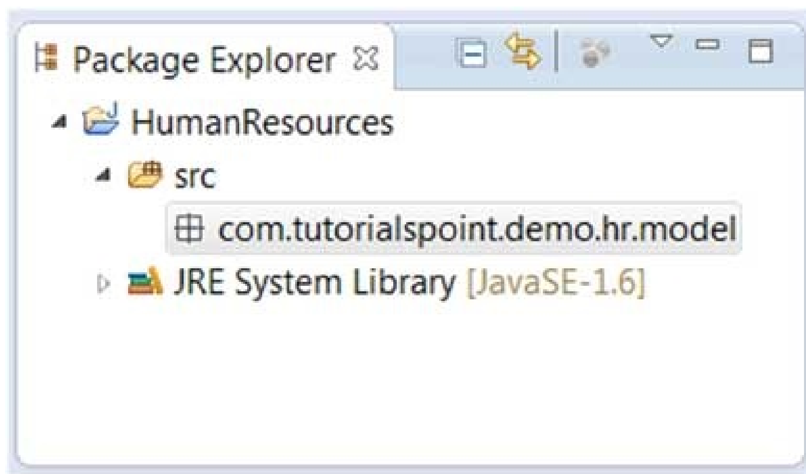
在向导弹出框(New Java Package)中可以执行以下操作：

- 输入资源文件夹名(Source Folder 字段)
- 输入包名(Name 字段)
- 点击 "Finish"按钮



查看新建包

在 Package Explorer 的资源文件夹下我们可以查看到新建的包。


Package Explorer
HumanResources
src
com.tutorialspoint.demo.hr.model
JRE System Library [JavaSE-1.6]

Eclipse 创建 Java 类

打开新建 Java 类向导

你可以使用新建 Java 类向导来创建 Java 类，可以通过以下途径打开 Java 类向导：

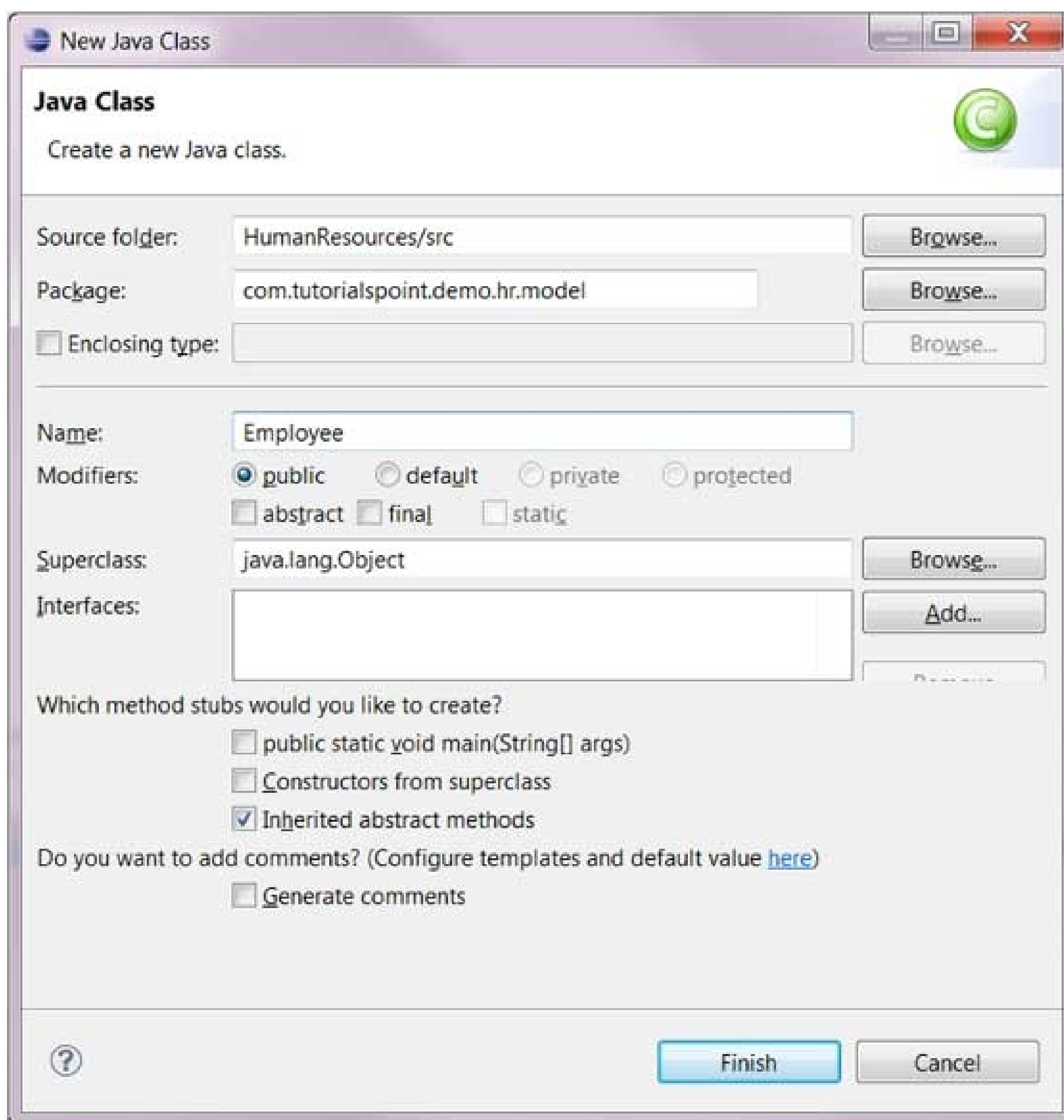
- 点击 "File" 菜单并选择 New > Class
- 在 Package Explorer 窗口中右击鼠标并选择 New > Class
- 点击类的下拉按钮 ( ) 并选择 ( )

在打开创建 Java 类向导前，最好选择好Java类所属的包名，这样在创建 Java 类时包名字段就会自动填充。

使用新建 Java 类向导

Java 类向导的弹窗中你可以进行以下操作：

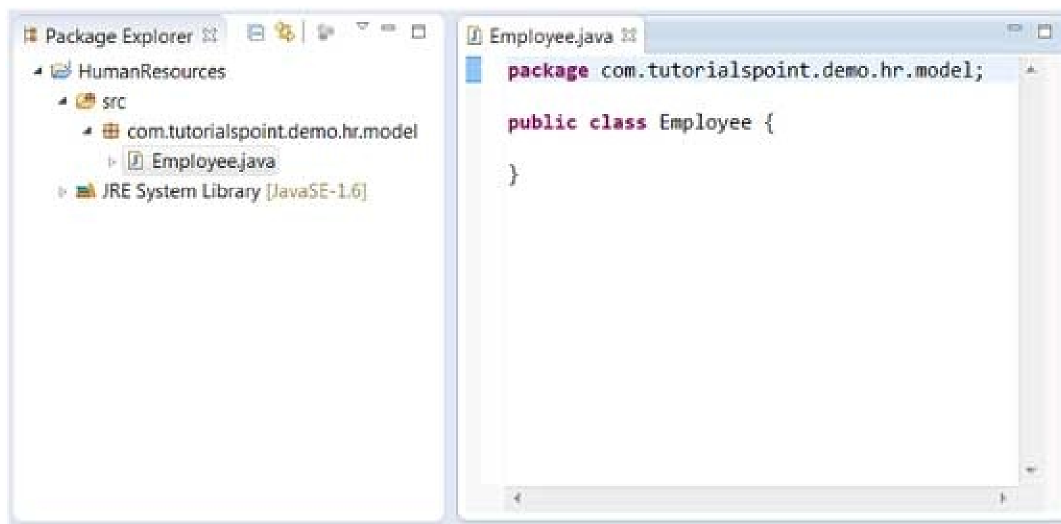
- 确认文件夹名(Source Folder)和包名(Package)是否正确
- 输入类名
- 选取其他修饰类
- 输入超类（Superclass）的名称或点击 Browse(浏览)按钮选择已存在的类
- 点击 Add(添加) 按钮选择类实现的接口
- 在复选框中可以选择方法创建方式及是否自动生成注释



- 点击 Finish(完成)按钮

查看新建的 Java 类

在 Package Explorer 视图中我们可以看到新建的类，我们可以通过右边的Java编辑器修改代码。


Package Explorer
HumanResources
src
com.tutorialspoint.demo.hr.model
Employee.java
JRE System Library [JavaSE-1.6]
Employee.java
package com.tutorialspoint.demo.hr.model;
public class Employee {
}

Eclipse 创建 Java 接口

打开新建 Java 接口向导

新建 Java 接口向导可以创建新的 Java 接口。打开向导的方式有：

- 点击 File 菜单并选择 New > Interface
- 在 Package Explorer 窗口中右击鼠标并选择 New > Interface
- 在工具条上的下拉框按钮中 ( ) 选择 ( )

在打开创建 Java 接口向导前，最好选择好Java接口所属的包名，这样在创建 Java 接口时包名字段就会自动填充。

使用新建 Java 接口向导

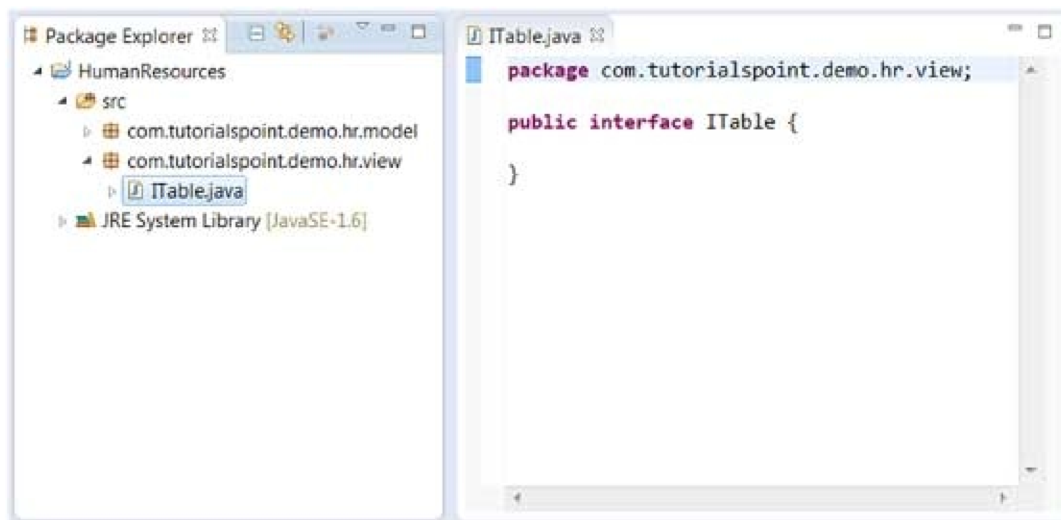
Java 接口向导的弹窗中你可以进行以下操作：

- 确认文件夹名(Source Folder)和包名(Package)是否正确
- 输入接口名称
- 点击 Add(添加) 按钮并选择要接口，该接口将被继承
- 选择是否自动生成注释
- 点击 Finish(完成) 按钮



查看新建的 java 接口

在 Package Explorer 视图中我们可以看到新建的接口，我们可以通过右边的Java编辑器修改接口代码。


Package Explorer
HumanResources
src
com.tutorialspoint.demo.hr.model
com.tutorialspoint.demo.hr.view
ITable.java
JRE System Library [JavaSE-1.6]
ITable.java
package com.tutorialspoint.demo.hr.view;
public interface ITable {
}

Eclipse 创建 XML 文件

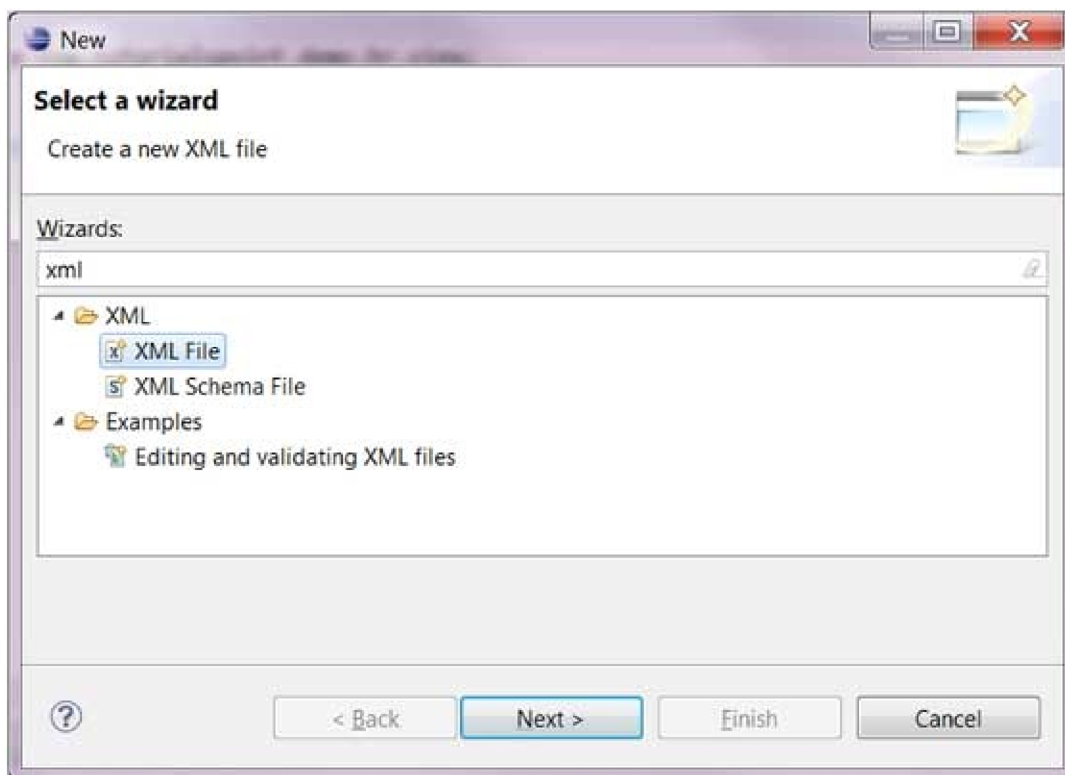
打开新建 XML 文件向导

你可以使用新建 XML 文件向导来创建 XML 文件。打开向导的方式有：

- 点击 File 菜单并选择 New > Other
- 点击新建下拉框 ( ) 选择 Other
- 快捷键组合： ctrl + N

在向导对话框中可以进行以下操作：

- 在输入框中输入 XML，会显示关联 XML 的向导
- 在展开的 XML 类别中选择 XML 文件



- 点击 Next 按钮进入新建 XML 文件向导

注意:

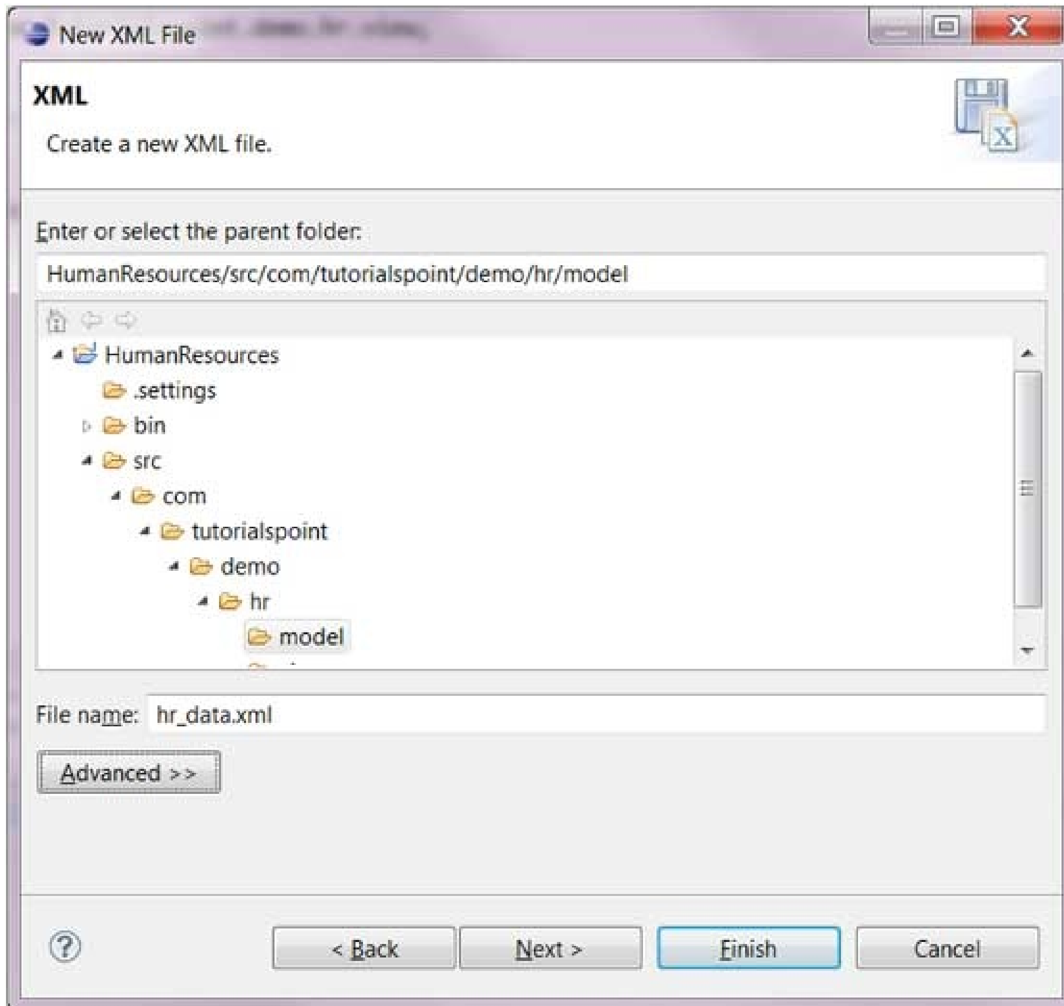
接下来在向导中我们还可以做以下操作：

- 点击 File 菜单并选择 New > XML File
- 在工具条上点击 XML File 按钮 (XML File...)

使用新建的 XML 文件向导

在新建 XML 文件向导中我们可以进行如下操作：

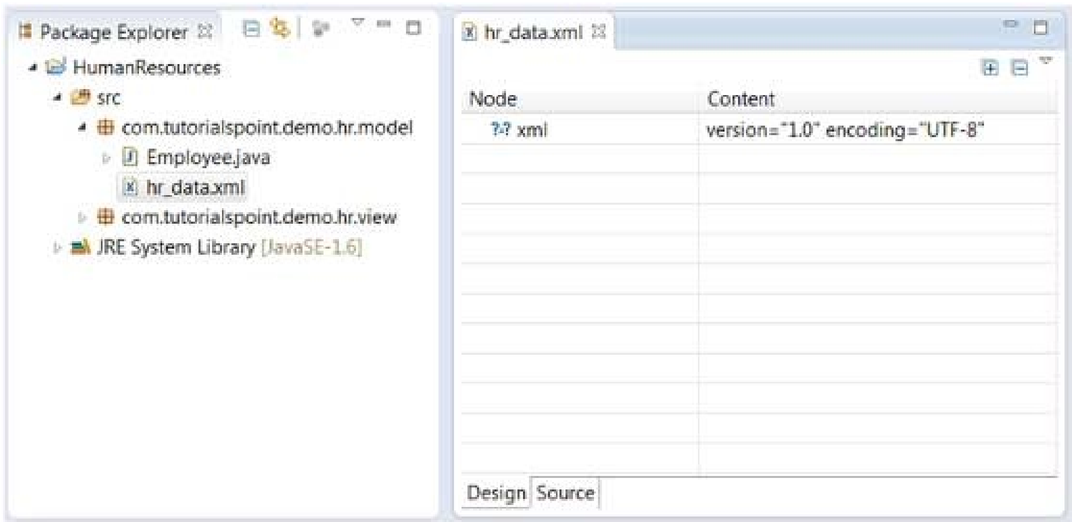
- 输入或选择 XML 文件所属的文件夹
- 输入 xml 文件名



- 点击 Next 按钮可以配置 DTD, XML Schema 的 XML 模式描述语言， 或者你可以直接点击 Finish 按钮完成 XML 文件的创建。

查看新建的 XML 文件

在 Package Explorer 视图中我们可以看到新建的 XML 文件，在右边的 XML 编辑器中我们可以修改新建的 XML 文件 。



XML 编辑器可以使用视图模式或源码模式来设计 XML 文件。

Eclipse Java 构建路径

设置 Java 构建路径

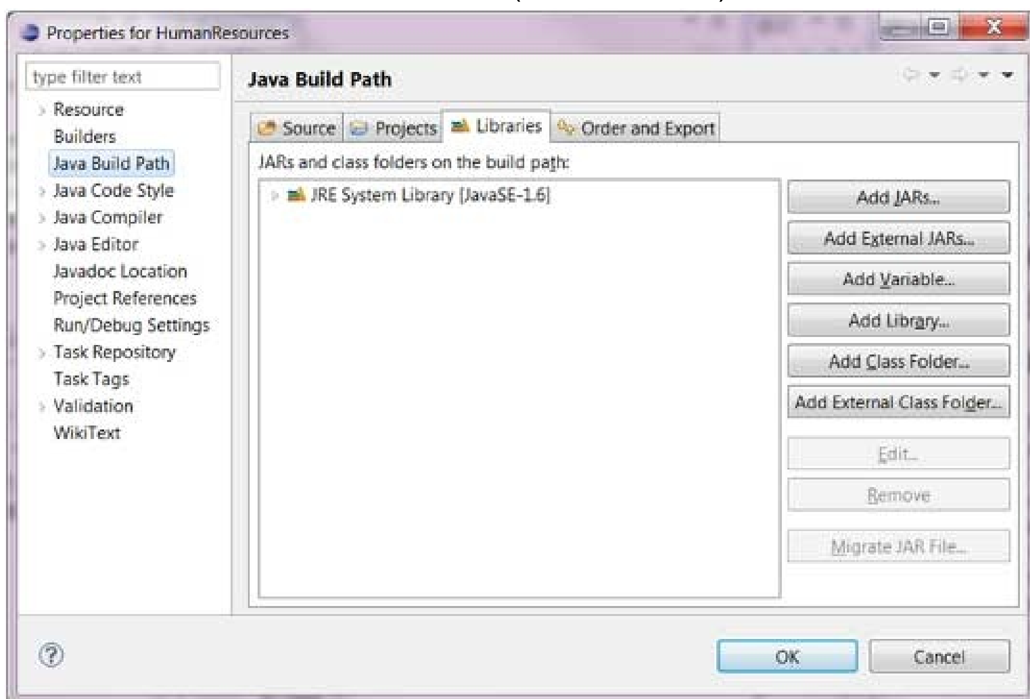
Java构建路径用于在编译Java项目时找到依赖的类，包括以下几项：

- 源码包
- 项目相关的 jar 包及类文件
- 项目引用的的类库

我们可以通过使用 Java 项目属性对话框中的 Java Build Path(Java 构建路径)选项来查看和修改 Java 构建路径。

Java 项目属性对话框可以通过在 Package Explorer 视图中鼠标右击指定的 Java 项目并选择 Properties(属性) 菜单项来调用。

然后 在左边窗口选择 Java Build Path(Java 构建路径)。



在 Java 构建路径窗口中我们可以已经引用到的 jar 包。

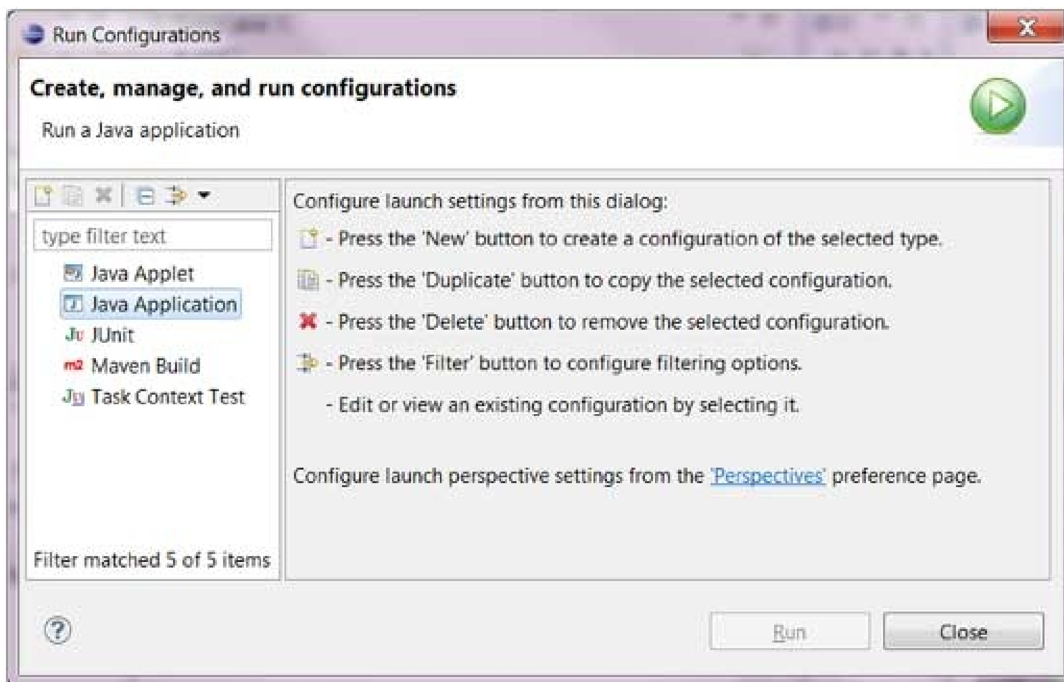
引用 jar 包可以在 Libraries 选项卡中完成，在 Libraries 选项卡中我们可以通过点击 Add JARs 来添加 Eclipse 工作空间中存在的jar包或 点击 External JARs 来引入其他文件中的 jar 包。

Eclipse 运行配置(Run Configuration)

创建和使用 Eclipse 运行配置

在运行配置(Run Configuration)对话框中可以创建多个运行配置。每个配置可以在应用中启用。

运行配置(Run Configuration)对话框可以通过 Run 菜单中选择 Run Configurations 来调用。



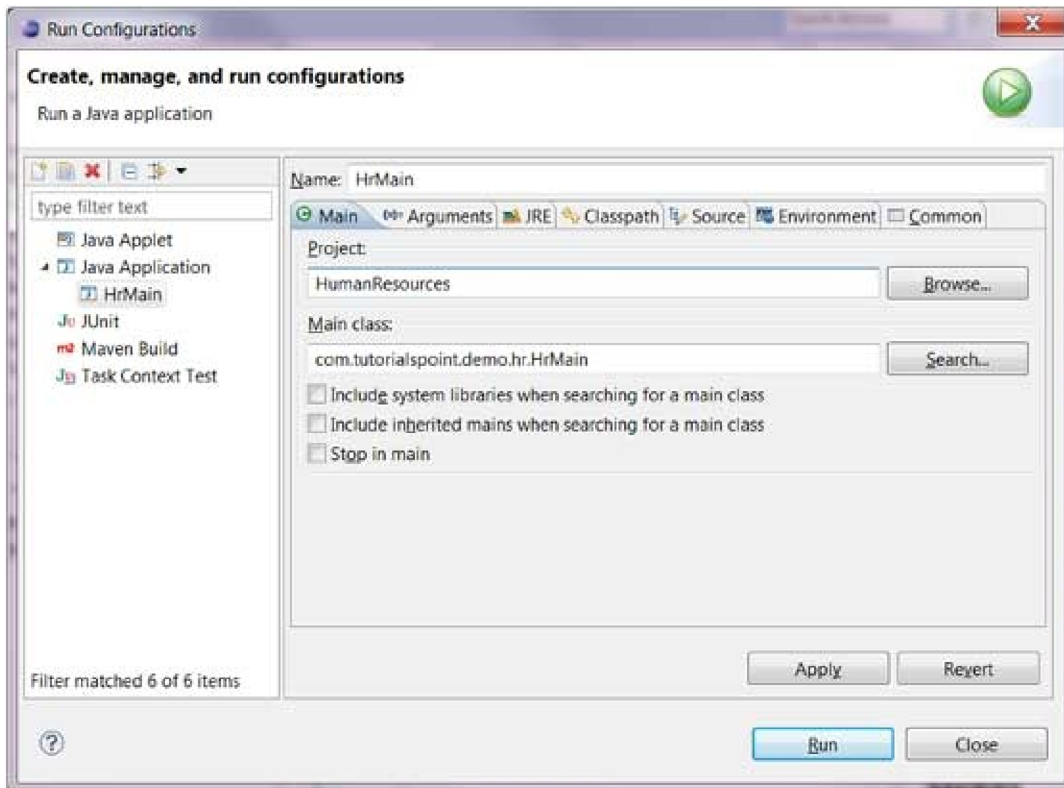
如果要给 Java 应用创建运行配置需要在左侧列表中选择 "Java Application" 并点击 New 按钮。

对话框中描述的项有：

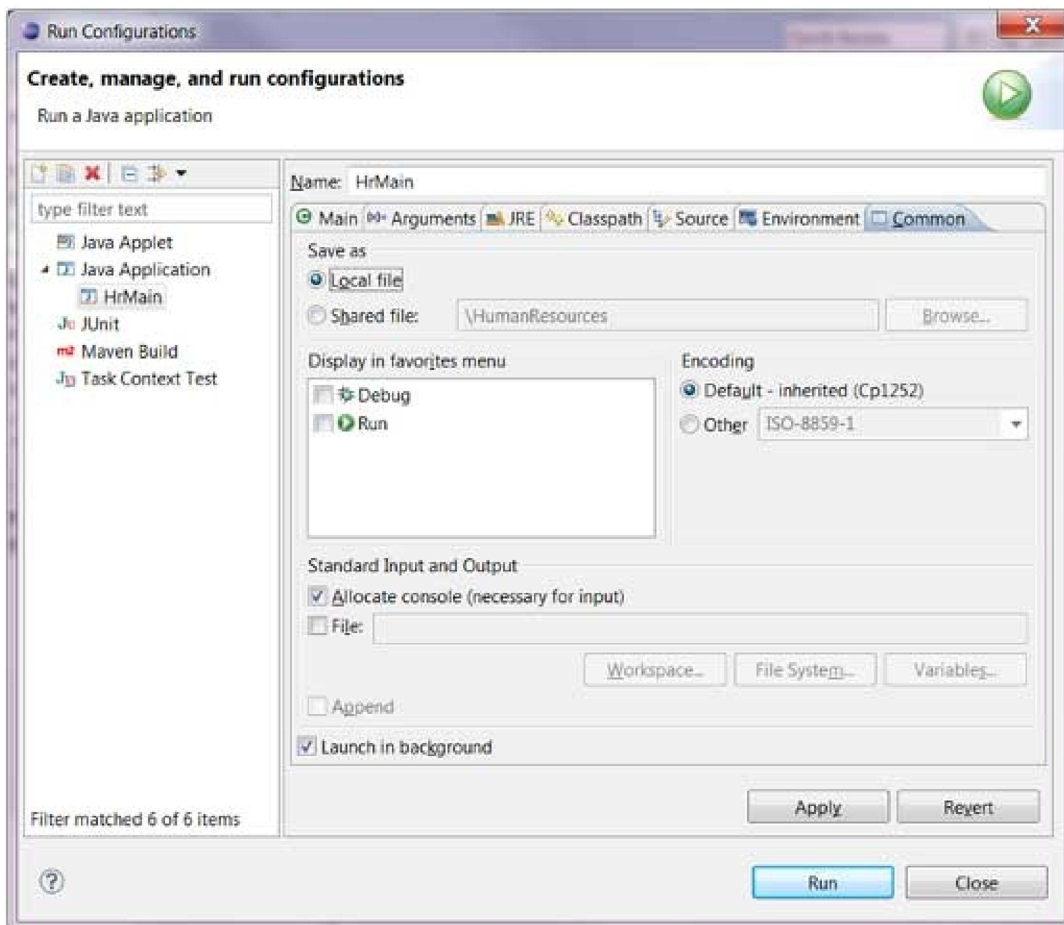
- 运行配置名称
- 项目名
- 主类名

Arguments(参数)项有：

- Program arguments(程序参数) 可以 0 个或多个
- VM arguments(Virtual Machine arguments:虚拟机参数) 可以 0 个或多个



Commons 选项卡中提供了通用配置，如标准输入输出的选项，可以到控制台或指定文件。



点击 Apply(提交) 按钮保存运行配置并点击 Run(运行) 按钮重新执行 Java 应用。

Eclipse 运行程序

运行 Java 程序

我们可以在 Package Explorer 视图

可以在 Package Explorer 视图中快速运行 Java 程序。

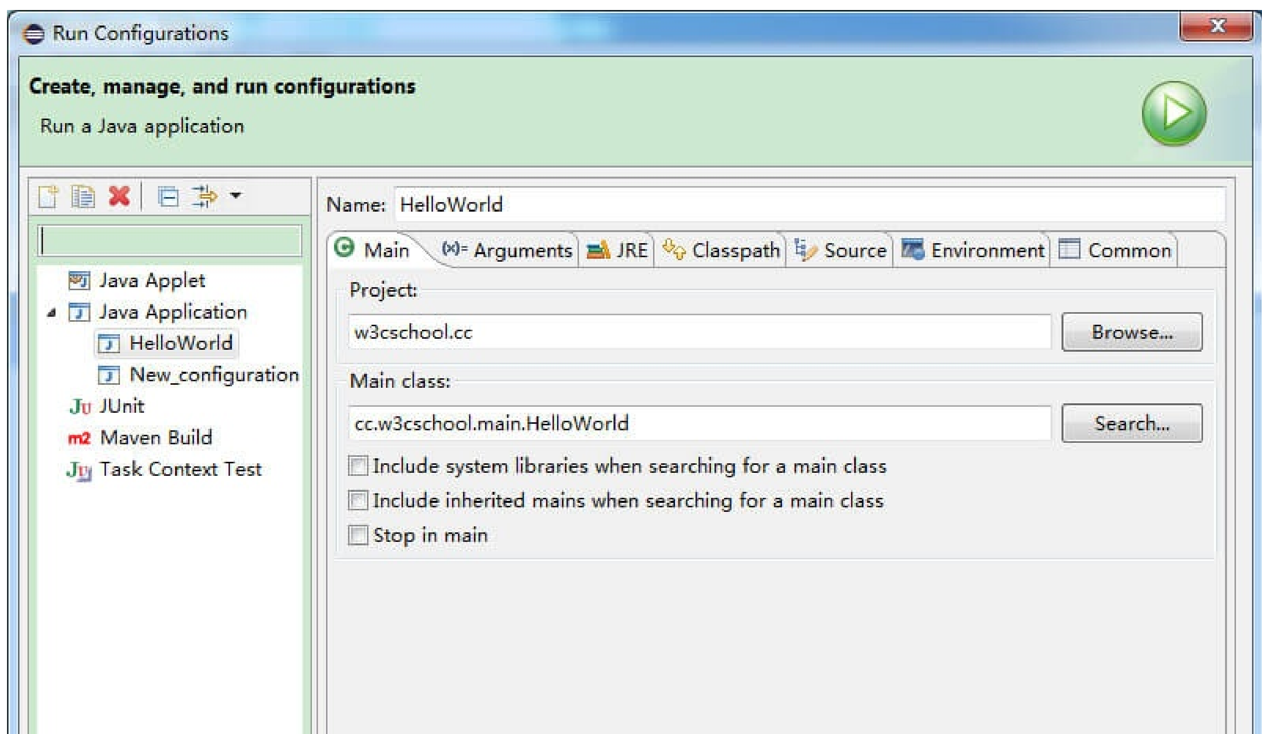
Package Explorer 视图:

鼠标右击包好 main 函数的 java 类选择 Run As > Java Application

同样你也可以在 Package Explorer 视图中选择包含 main 方法的类并按下快捷键： Alt + Shift + X, J

以下两种方式都能创建一个新的 Run Configuration(运行配置) 我们可以使用它来启动 Java 应用程序。

如果运行配置已经创建，你可以在 Run 菜单中选择 Run Configurations 来启动 Java应用，点击运行配置的名称，然后点击运行按钮的Java应用程序。



Run 菜单中的 Run 选项可以重新启动先前启动 Java 应用。



重新启动先前启动 Java 应用快捷键为 Ctrl + F11。

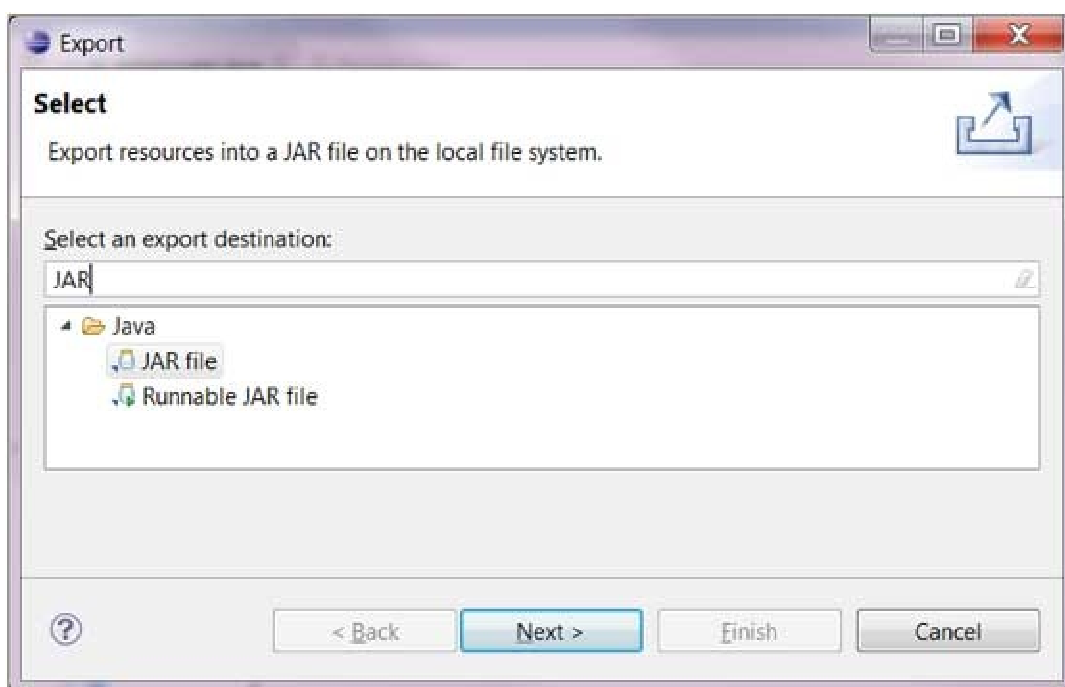
Eclipse 生成jar包

打开 Jar 文件向导

Jar 文件向导可用于将项目导出为可运行的 jar 包。

打开向导的步骤为:

- 在 Package Explorer 中选择你要导出的项目内容。如果你要导出项目中所有的类和资源，只需选择整个项目即可。
- 点击 File 菜单并选择 Export。
- 在输入框中输入"JAR" 。

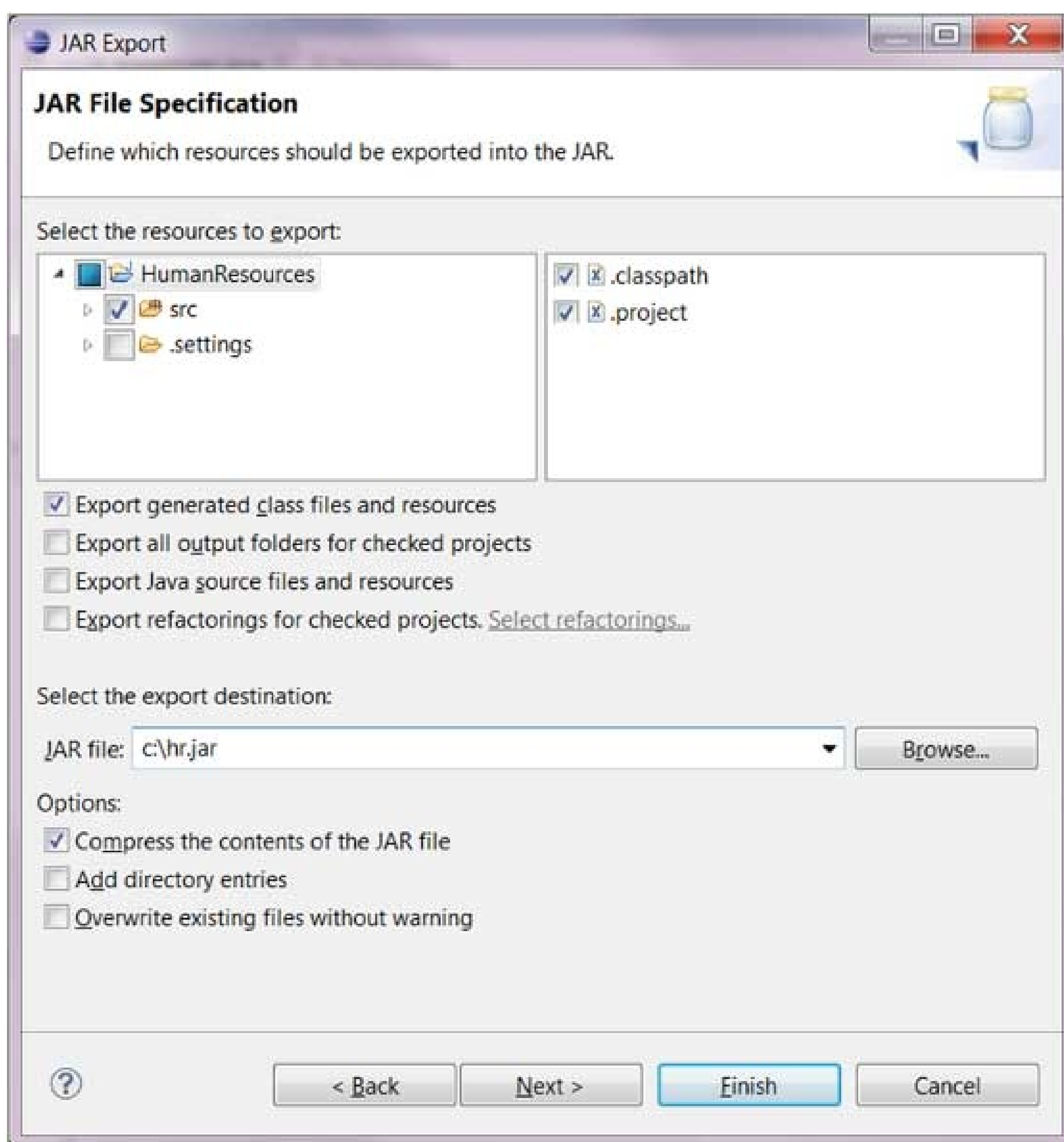


- 在选项中选择 JAR file 选项开启向导。
- 点击 Next 按钮

使用 Jar 文件向导

在 JAR File Specification（JAR 文件描述） 页面可进行以下操作：

- 输入 JAR 文件名及文件夹
- 默认只导出 class 文件。你也可以通过勾选 "Export Java source files and resources" 选项来导出源码的内容。



- 点击 Next 按钮修改 JAR 包选项
- 点击 Next 按钮修改 JAR Manifest 描述信息
- 点击 Finish 按钮完成操作

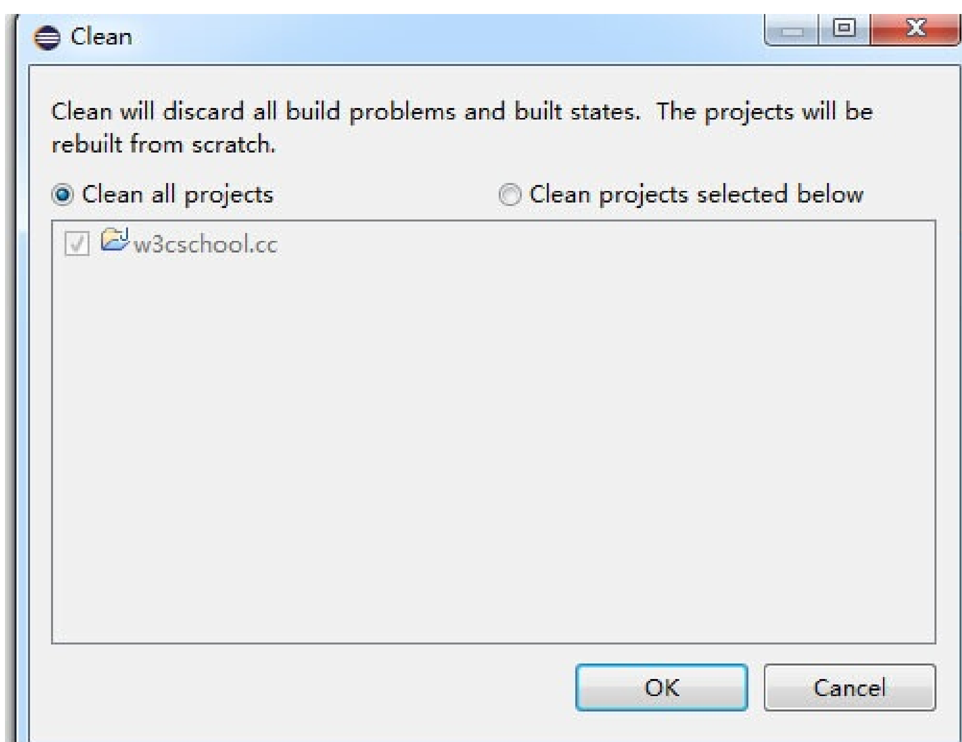
Eclipse 关闭项目

为什么要关闭项目？

Eclipse 工作空间包含了多个项目。一个项目可以是关闭或开启状态。

项目打开过多影响有：

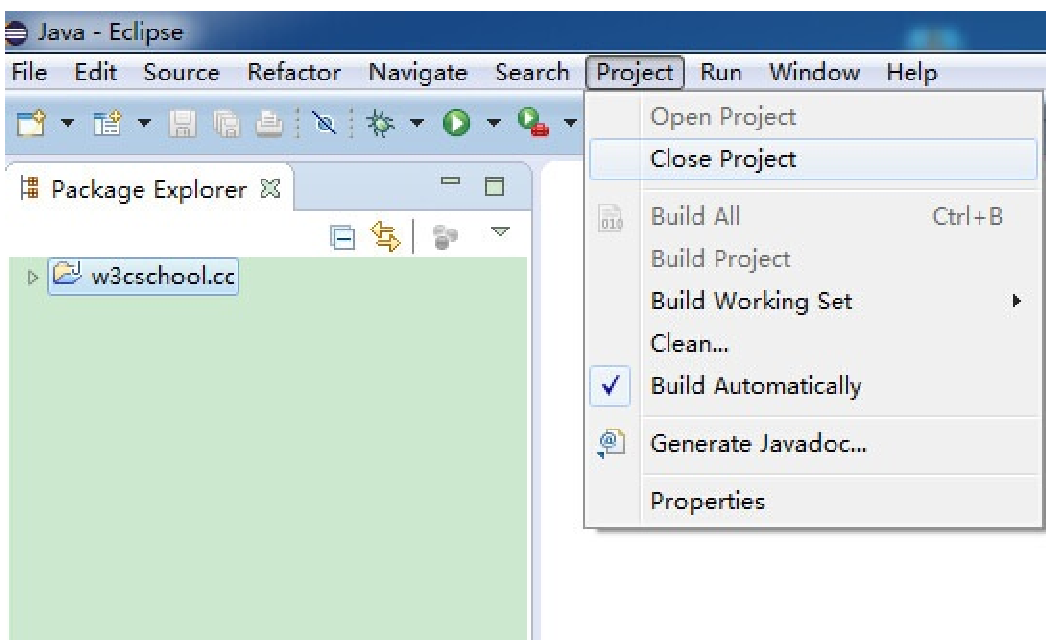
- 消耗内存
- 占用编译时间：在删除项目.class 文件（Clean All Projects）时并重新编译（在菜单上选择 Project > Clean > Clean all projects ）。



如何关闭项目?

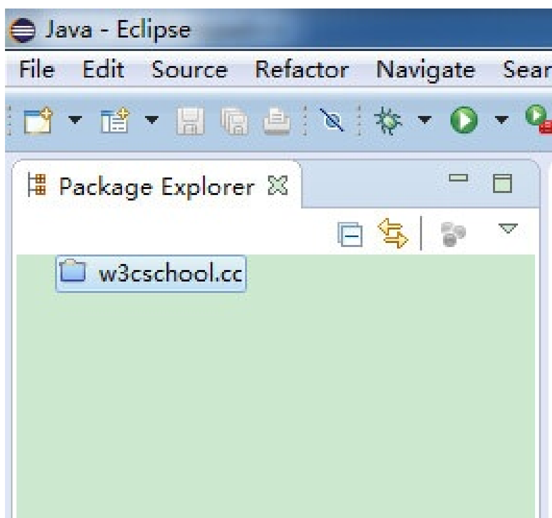
如果项目不处于开发阶段，我们就可以先关闭项目。

在 Package Explode 视图上选择要关闭的项目，并通过菜单上选择 Project > Close Project 来关闭项目。



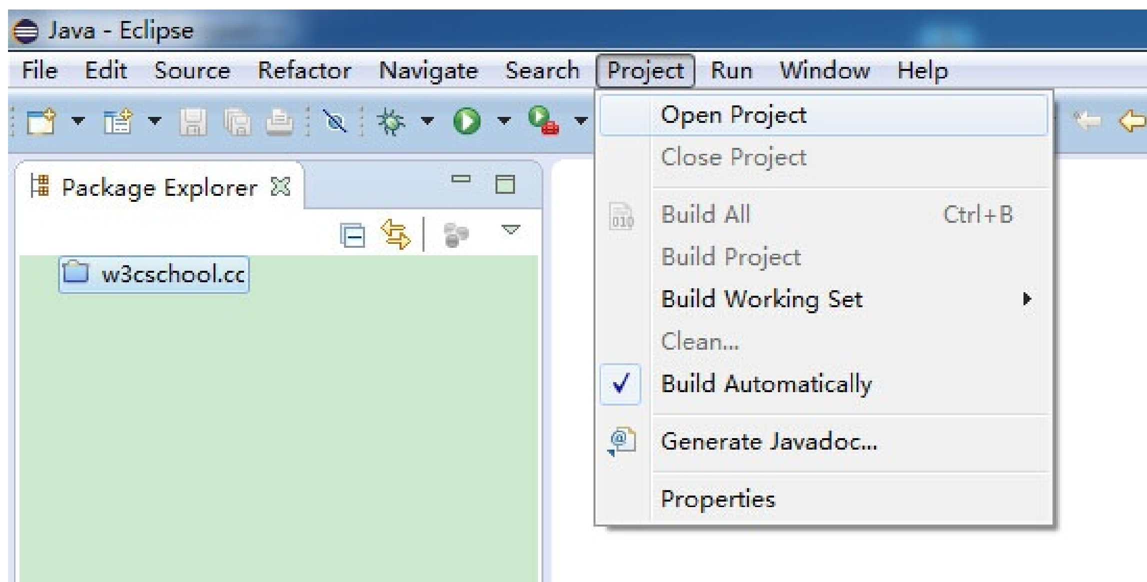
关闭后的项目

项目关闭后我们可以在 Package Explorer 视图看到项目的图标已经变了。关闭后的项目是不能编辑的。



重新开启项目

你可以通过选择 Project > Open Project。


Java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
w3cschool.cc
Open Project
Close Project
Build All Ctrl+B
Build Project
Build Working Set
Clean...
Build Automatically
Generate Javadoc...
Properties

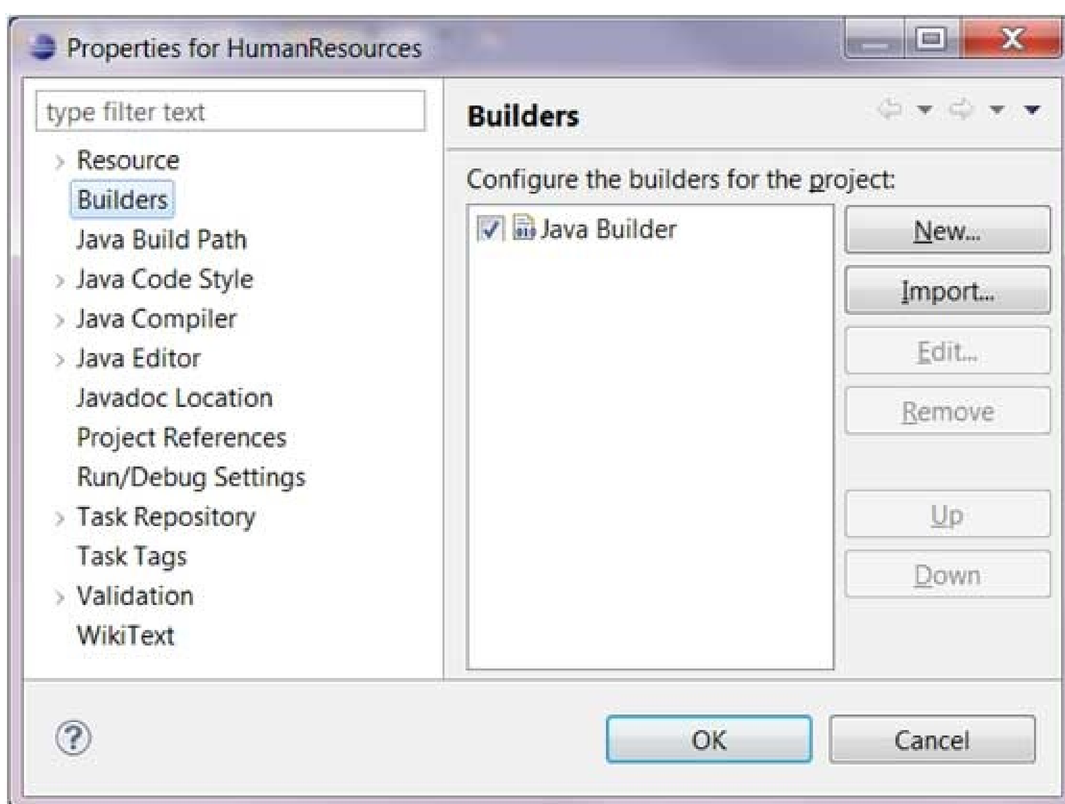
Eclipse 编译项目

编译 Java 项目

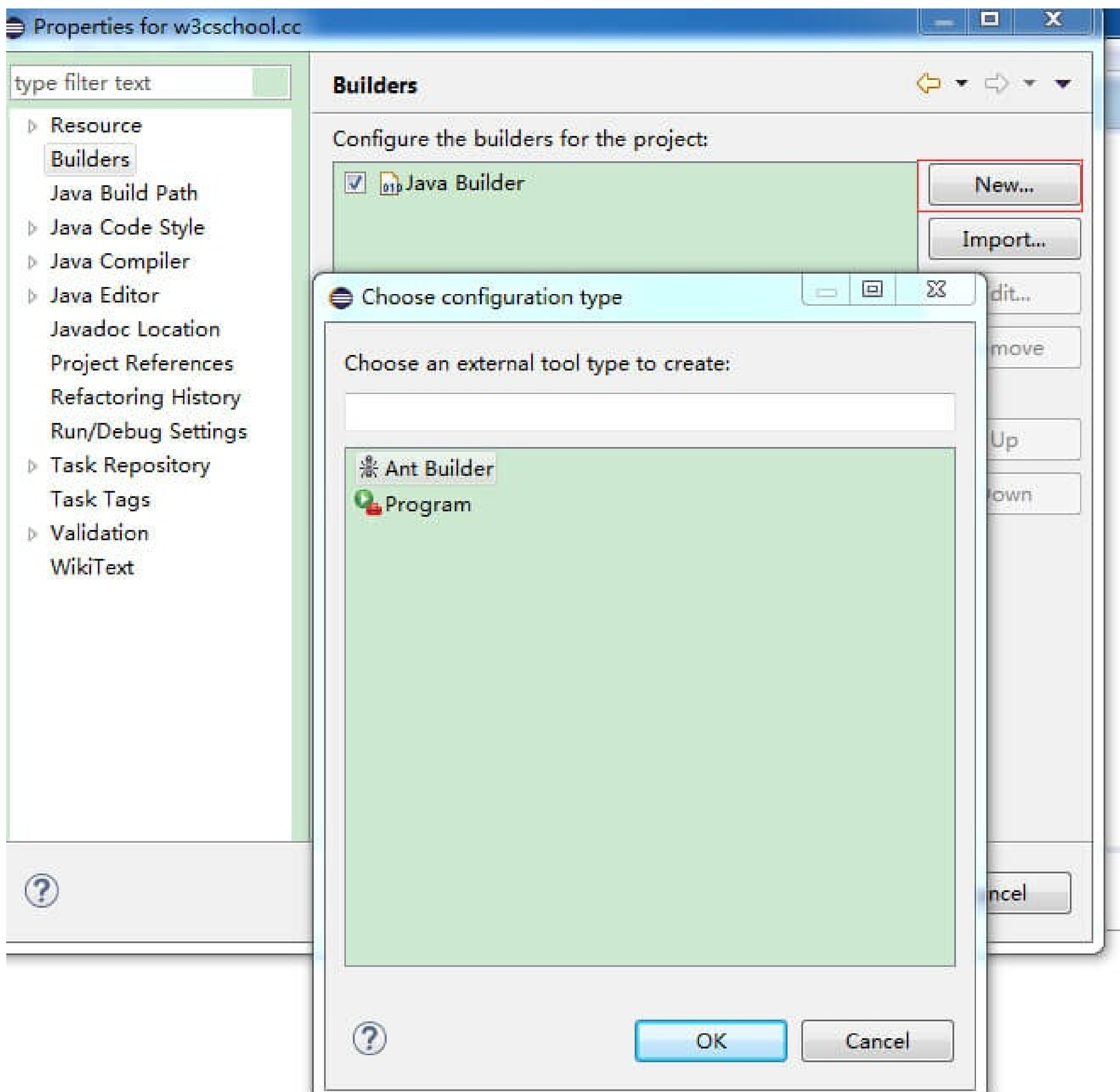
一个项目可以关联多个编译器。

java 项目关联的是 java 编译器。可以通过以下方式来查看项目关联的编译器：

- 在 Package Explorer 视图中鼠标右击项目并选择 Properties
- 在左侧的树形菜单中点击 Builders

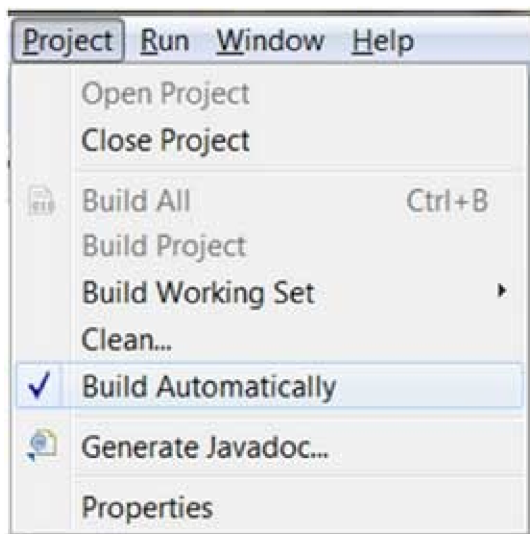


java编译器用于编译java项目。通过点 New 按钮我们可以让java项目关联 Ant builder 编译器。



java 编译器通过编译 java 项目生成 class 文件。当项目源码发生变化时会自动重新编译 java 代码。

可以通过去除 Project 菜单中 Build Automatically (自动编译)项来禁用自动编译功能。



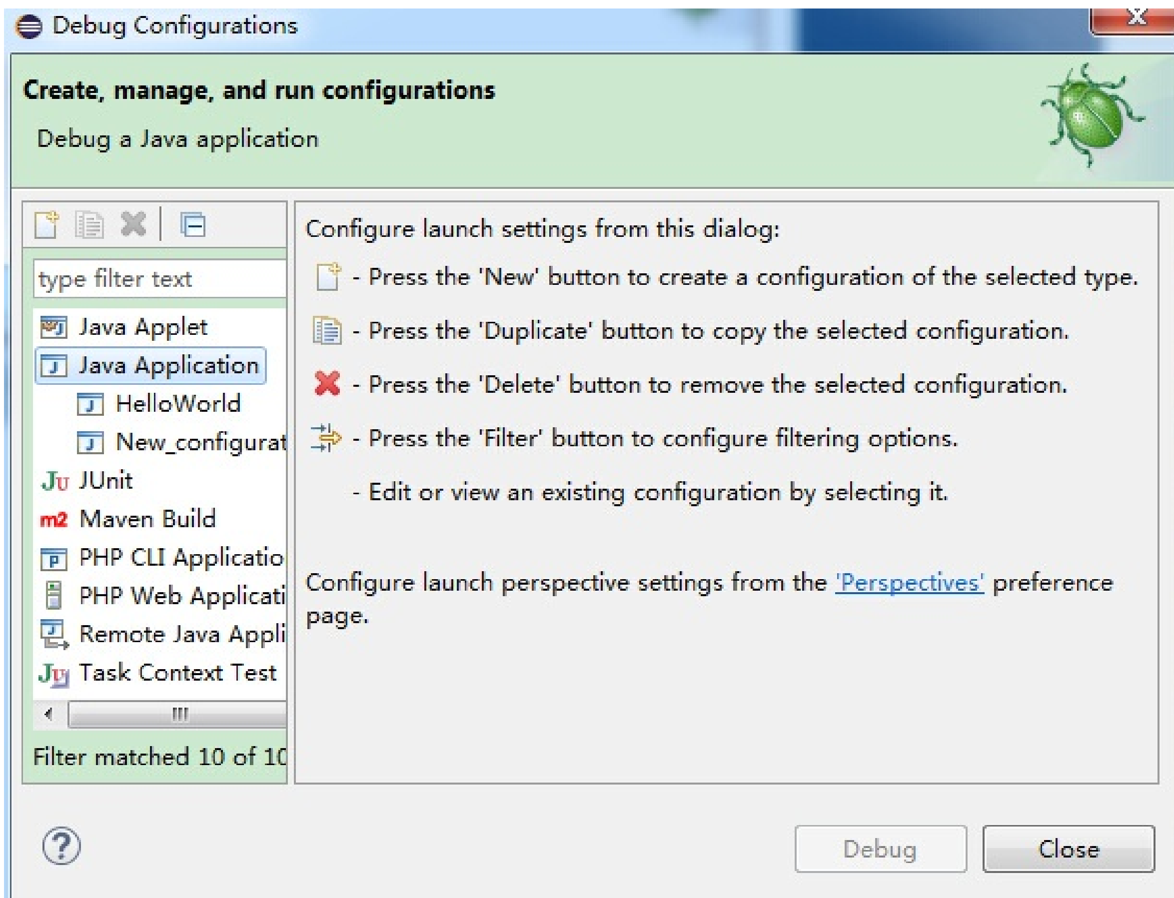
如果你禁用了自动编译功能，项目需要通过 Project 菜单中的 Build Project 菜单项来编译java 项目。 如果勾选了 Build Automatically(自动编译) 项，则 Build Project(手动编译) 菜单项是不可用的。

Eclipse Debug 配置

创建和使用 Debug 配置

Eclipse Debug 配置类似于运行配置但它是用于在调试模式下开启应用。

打开 Debug 配置对话框步骤为：Run > Debug Configurations 。



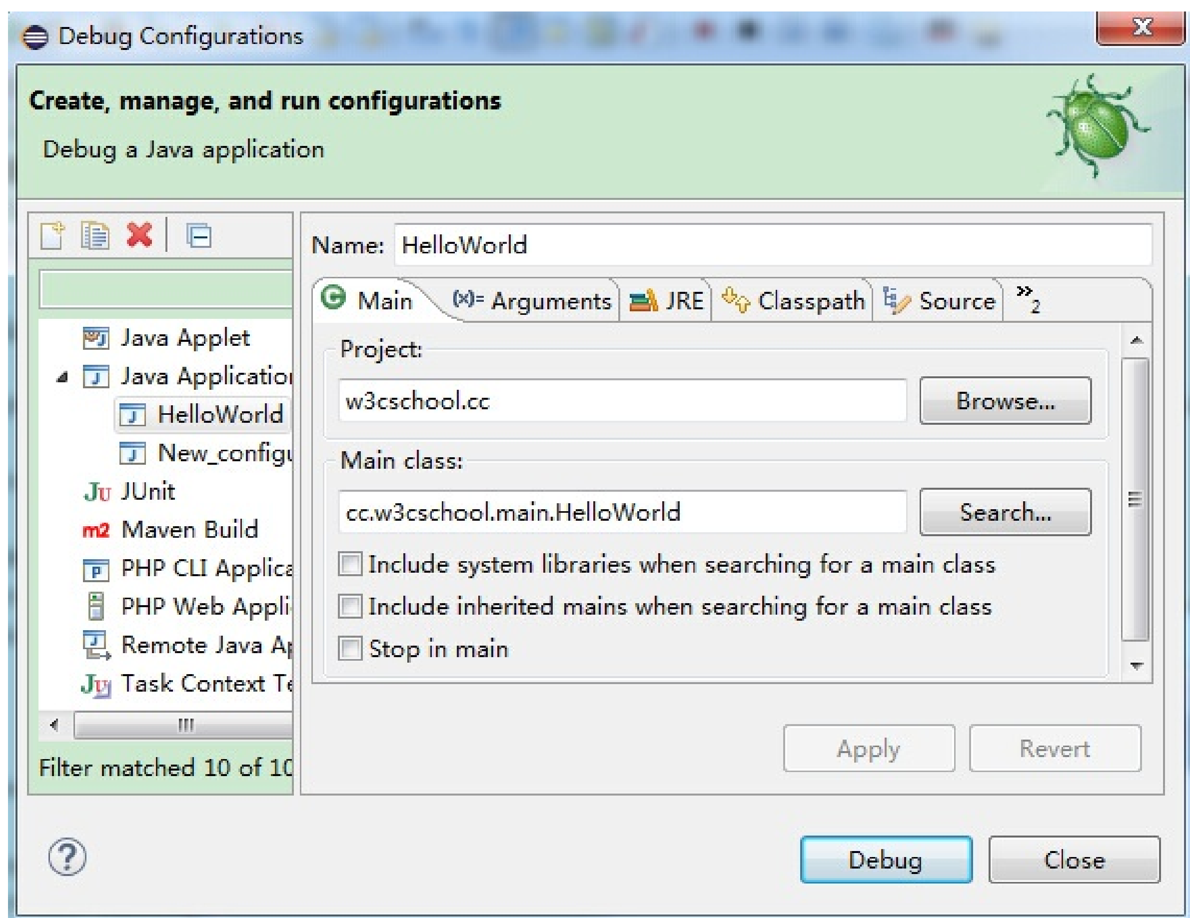
从左侧列表中选择 "Java Application" 选项来创建 Java 应用的调试配置并 New 按钮。

对话框中的描述信息有：

- 调试配置的名称
- 项目名称
- 主类名

arguments(参数)选项卡的描述信息有:

- 零个或多个程序参数
- 零个或多个虚拟机参数（VM arguments）



保存运行配置信息并点击 Apply 按钮，然后点击 Debug 按钮在调试模式下载入应用。

Eclipse Debug 调试

Debug 调试 Java 程序

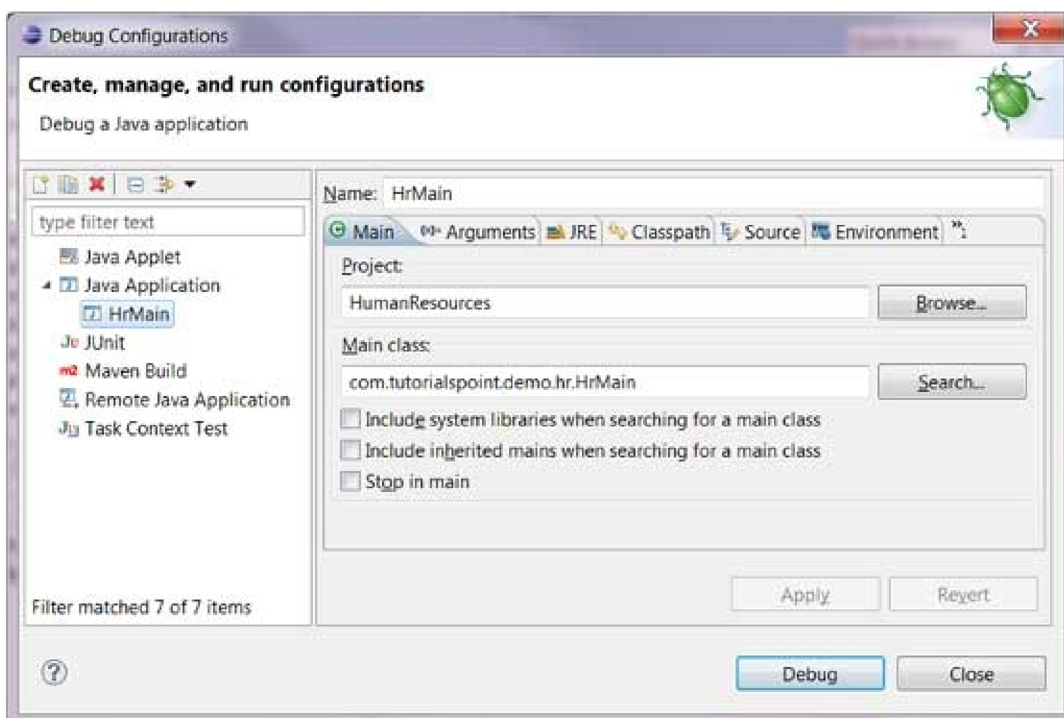
我们可以在 Package Explorer 视图调试 Java 程序，操作步骤如下：

- 鼠标右击包含 main 函数的 java 类
- 选择 Debug As > Java Application

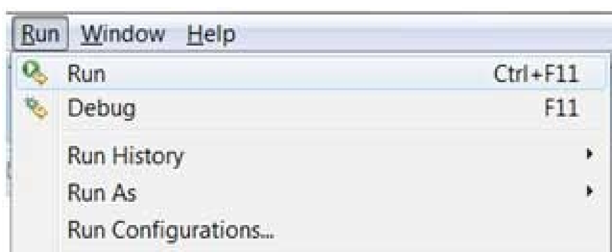
该操作也可以通过快捷键来完成，快捷键组合为 Alt + Shift + D, J。

以上操作会创建一个新的 Debug Configuration（调试配置），并使用该配置来启动 Java 应用。

如果 Debug Configuration（调试配置）已经创建，你可以通过 Run 菜单选择 Debug Configurations 选取对应的类并点击 Debug 按钮来启动 Java 应用。



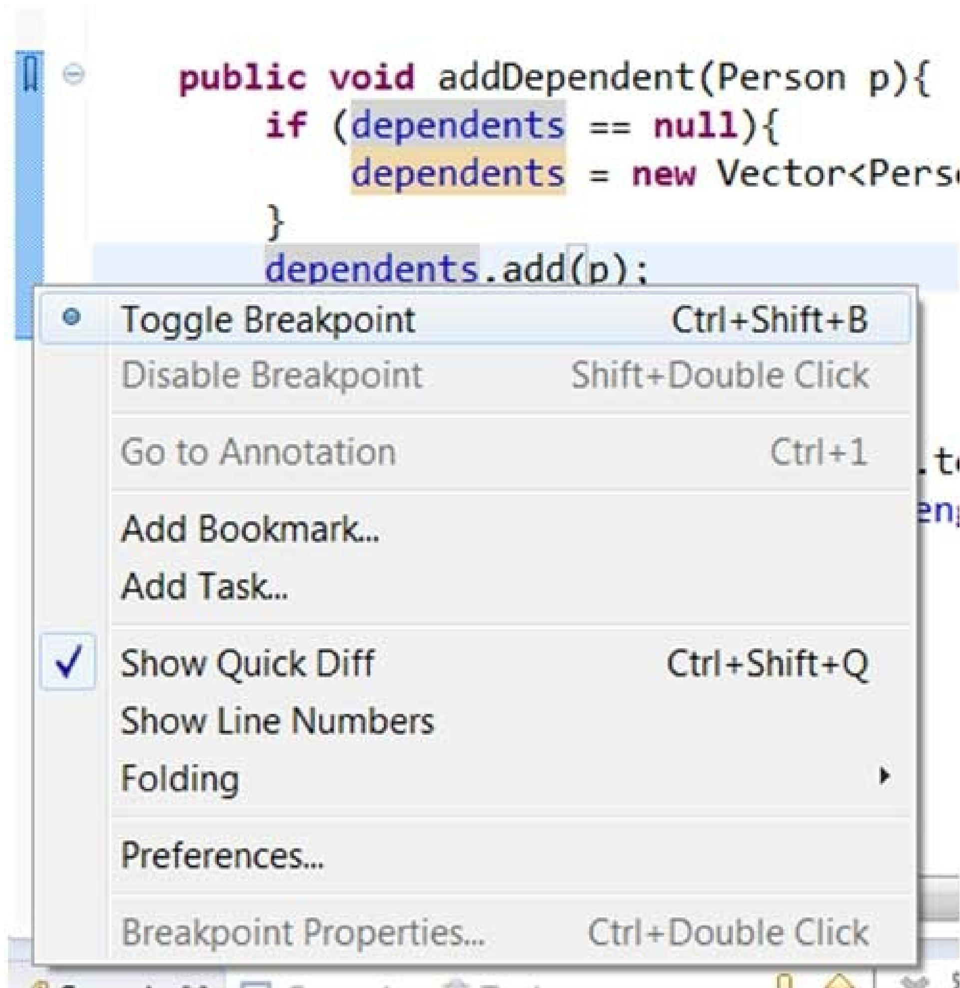
Run 菜单的 Debug 菜单项可以重新加载之前使用了调试模式的 java 应用。



重新加载之前使用了调试模式的 java 应用快捷键为 F11。

当使用调试模式开启java程序时，会提示用户切换到调试的透视图。调试透视图提供了其他的视图用于排查应用程序的故障。

java 编辑器可以设置断点调试。 在编辑器中右击标记栏并选择 Toggle Breakpoint 来设置断点调试。



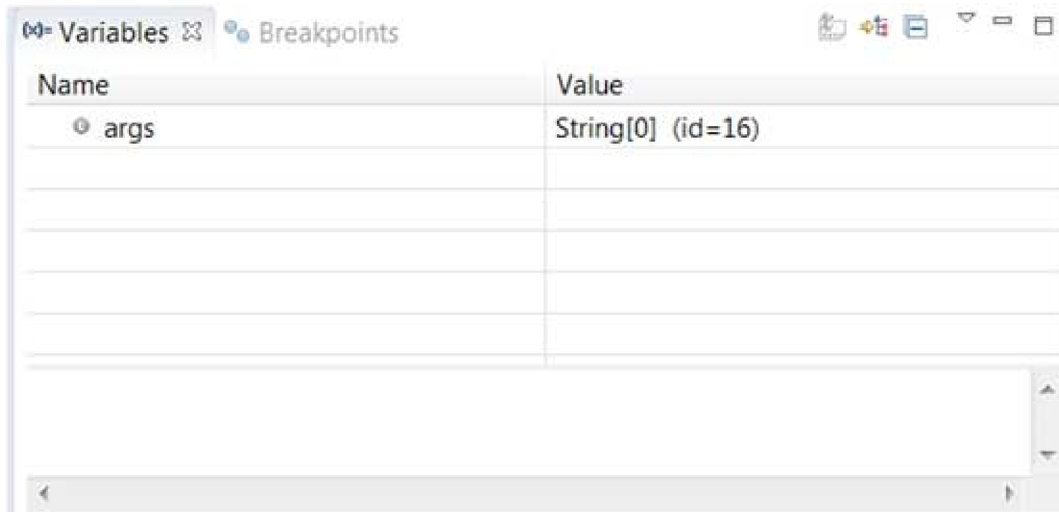
断点可以在标记栏中看到。也可以在 Breakpoints View（断点视图）中看到。

当程序执行到断点标记的代码时 JVM 会挂起程序，这时你可以查看内存使用情况及控制程序执行。

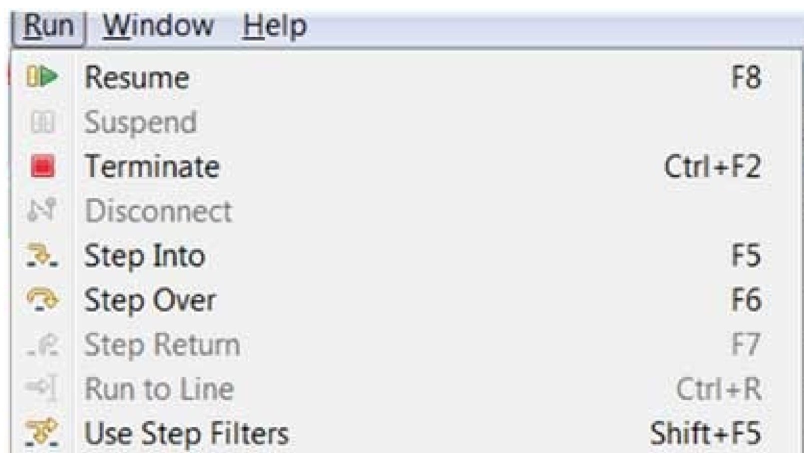
程序挂起时，Debug(调试)视图可以检查调用堆栈。



variables(变量)视图可以查看变量的值。



Run 菜单中有继续执行(Resume)菜单项，跳过(Step Over)一行代码，进入函数(Step Into)等。



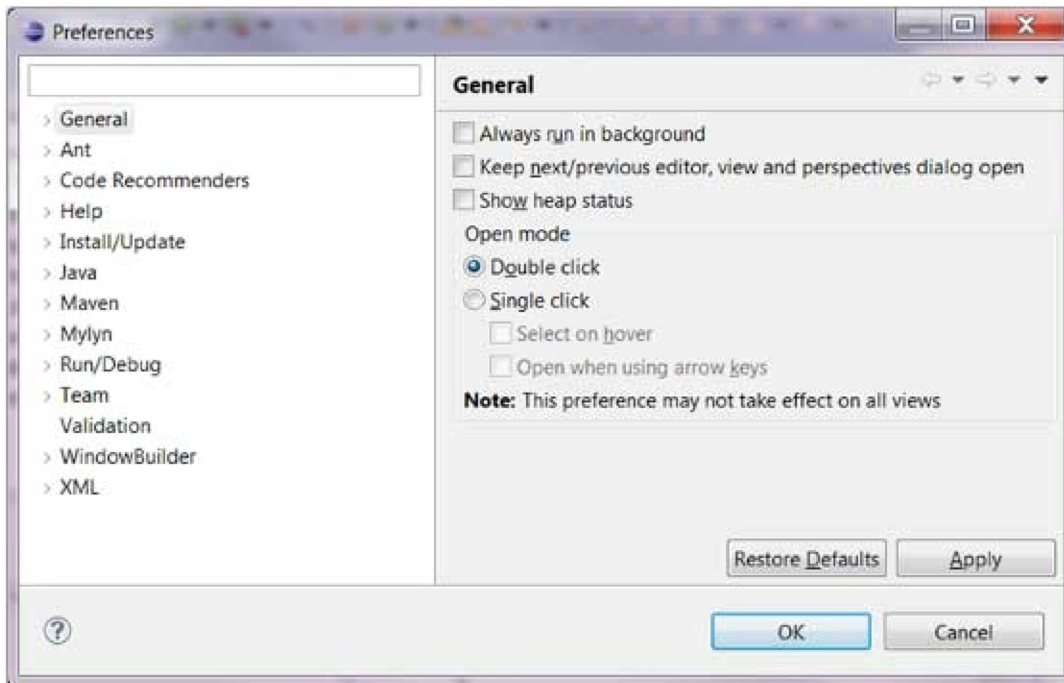
以上图片中显示了 Resume, Step Into 和 Step Over 等关联的快捷键操作。

Eclipse 首选项(Preferences)

设置首选项

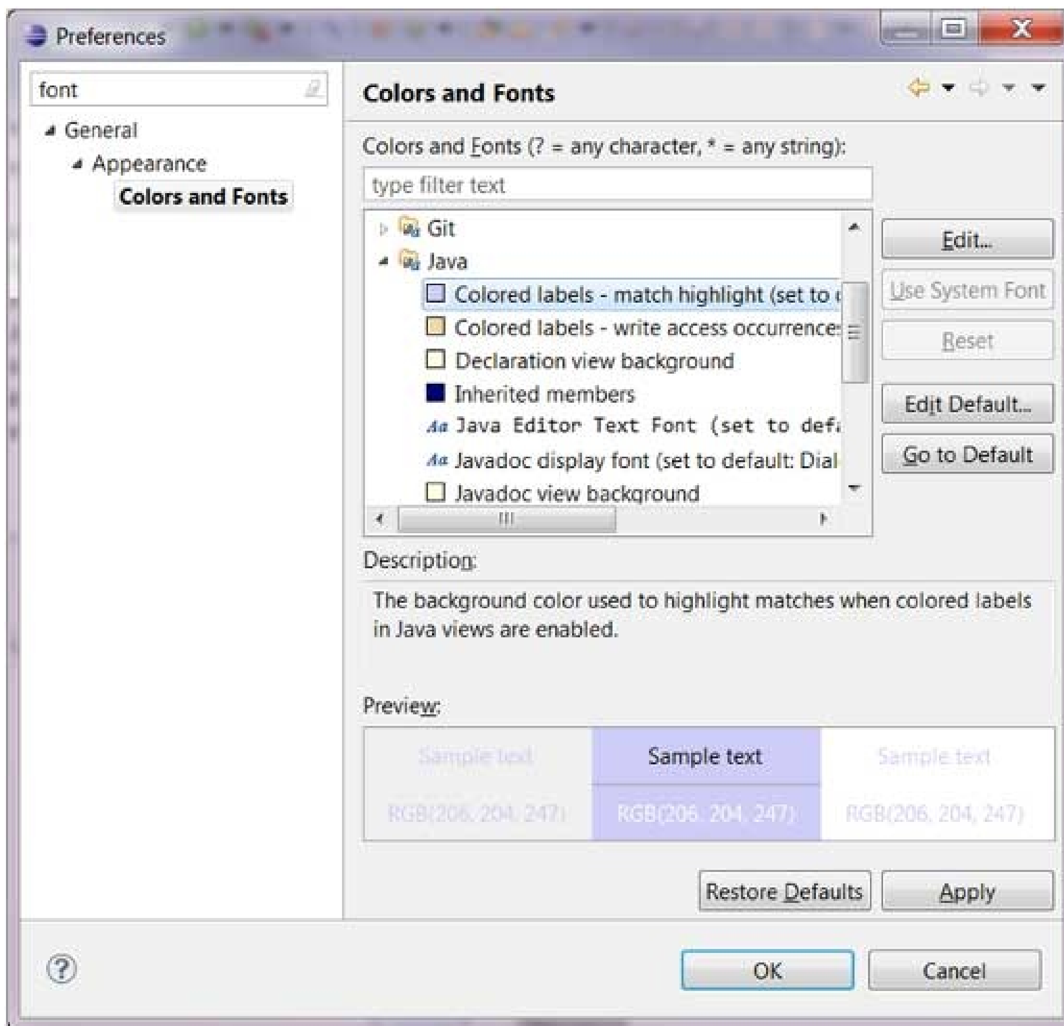
该对话框可通过框架管理但是其他插件可以设置其他页面来管理首选项的配置。

我们可以通过 Window 菜单选择 Preferences 菜单项来开启该对话框。



首选项页面有多个分类组成。你可以在左侧菜单中展开各个节点来查看首选项的配置。

左上角的输入框可以快速查找首选项页面。 你只需在输入框中输入要查找的首选项页面的字母即可快速找到对应的首选项页面。 例如：输入 font 即可查找到 Font(字体) 首选项页面。



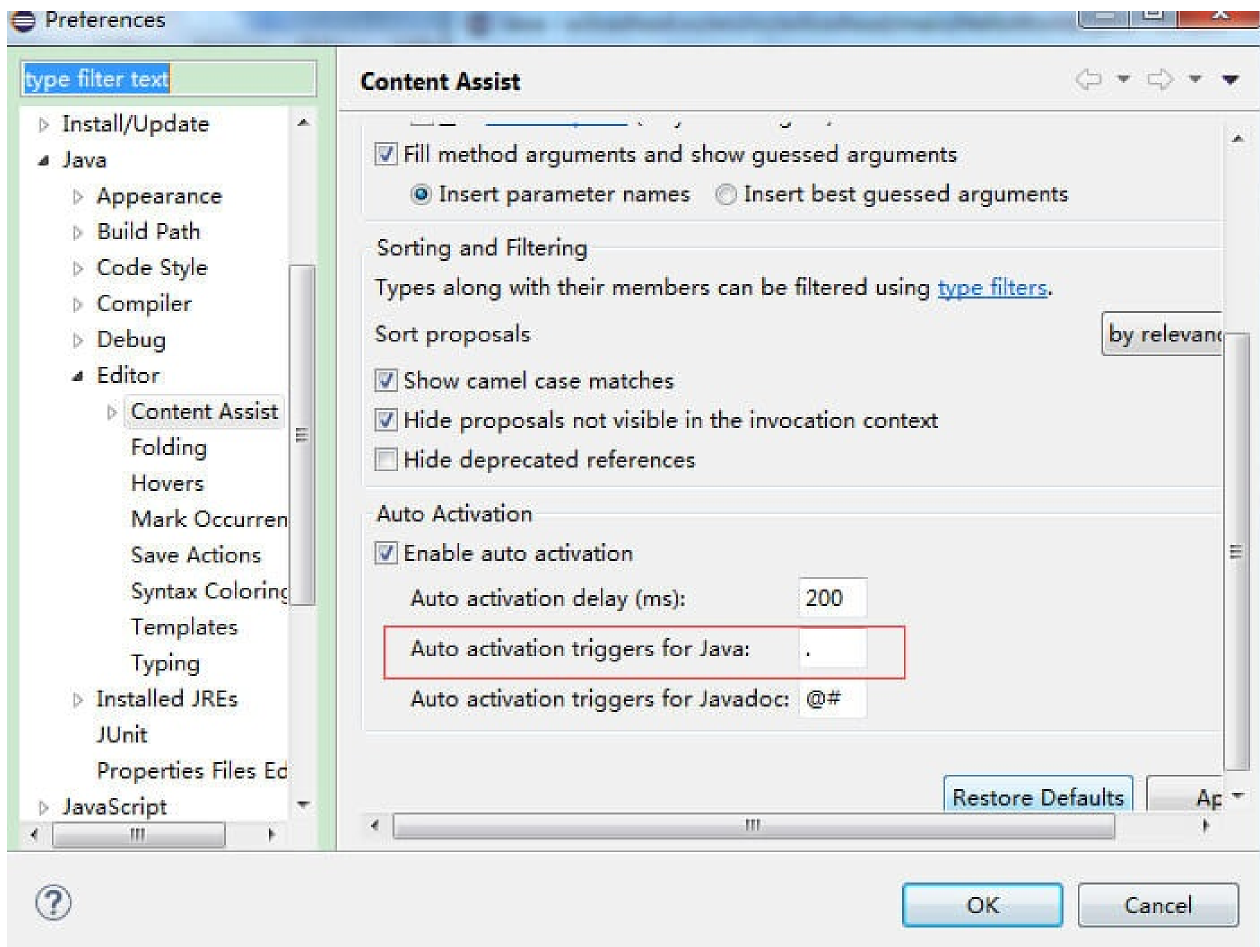
在你完成首选项页面的配置后点击 OK 按钮就可以保存配置，点击 Cancel 按钮用于放弃修改。

Eclipse 内容辅助

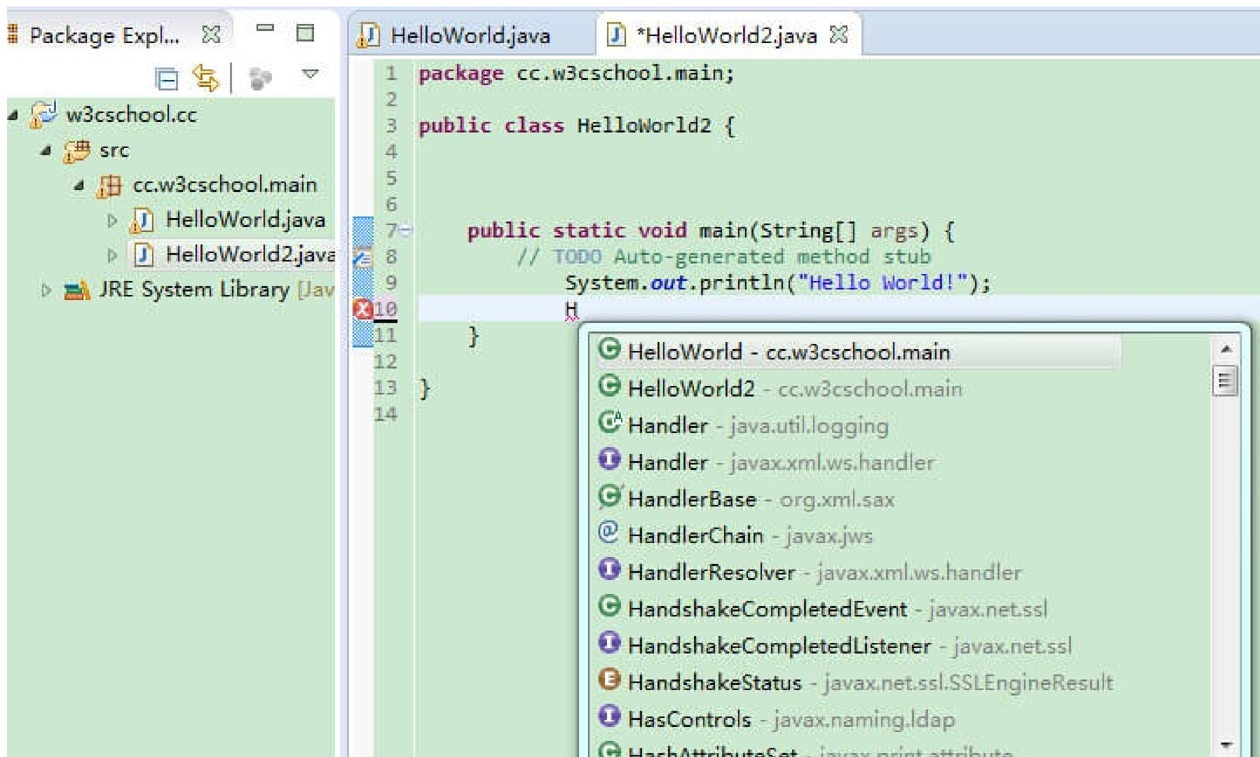
使用内容辅助

Eclipse中我们可以使用代码提示来加快开发速度，默认是输入"."后出现自动提示，用于类成员的自动提示。

设置自动提示的配置在：window->Preferences->Java->Editor->Content Assist：

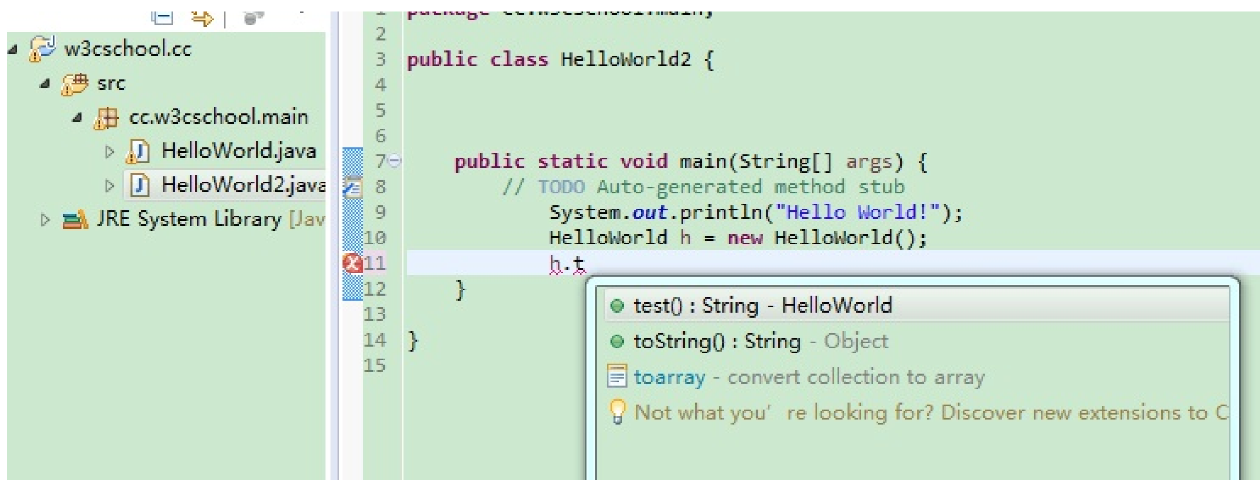


如果能在我们输入类的首字母按 **alt + /** 后就出现自动提示，。



输入 "." 后出现自动提示的内容有：

- 类变量
- 类方法
- 超类方法
- 其他相关类



Eclipse 快速修复

使用快速修复

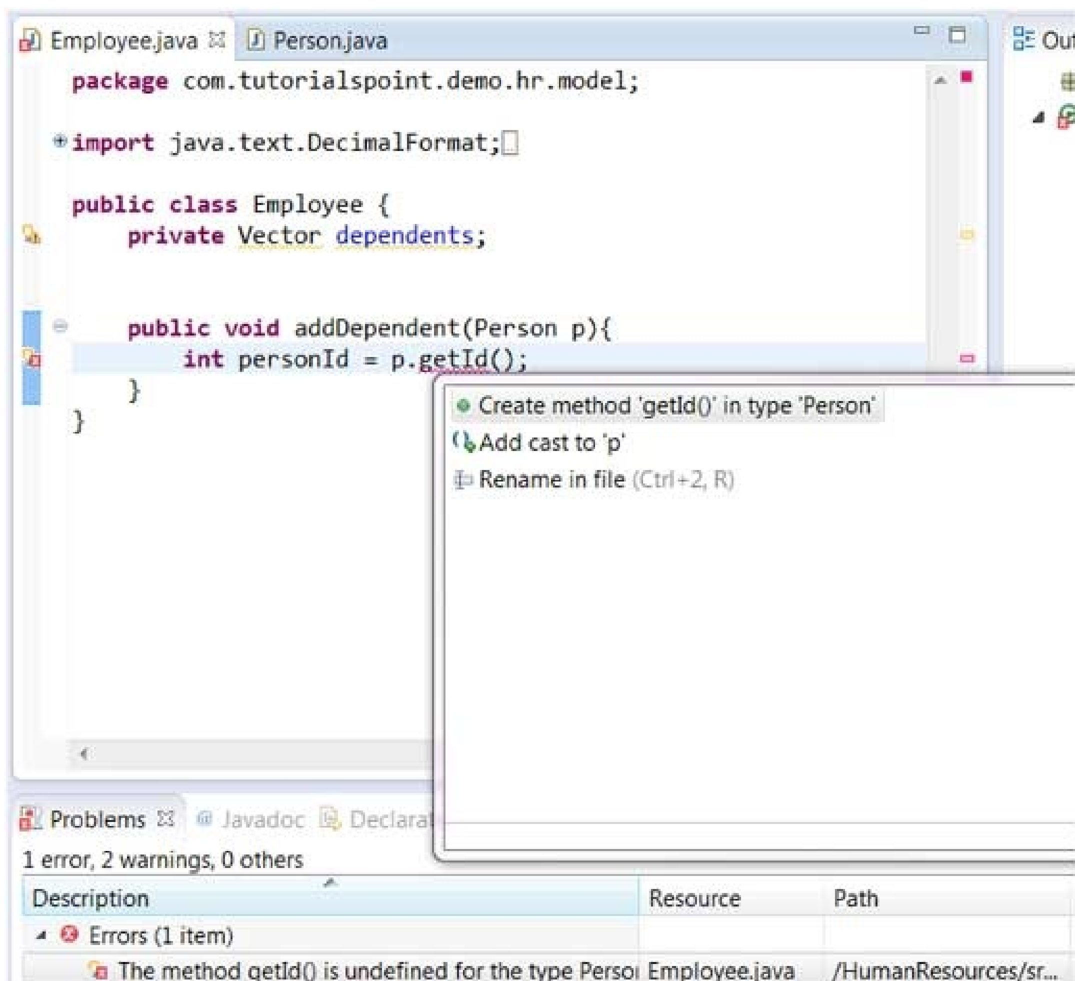
在 Eclipse 编辑器中当你输入字母时，编辑器会对你输入的内容进行错误分析。

Java 编辑器中使用 Java 语法来检测代码中的错误。当它发现错误或警告时：

- 使用红色波浪线突出错误
- 使用黄色的波浪线突出警告
- 在 Problem 视图中显示错误和警告
- 在垂直标尺上显示黄色小灯泡及警告和错误标识

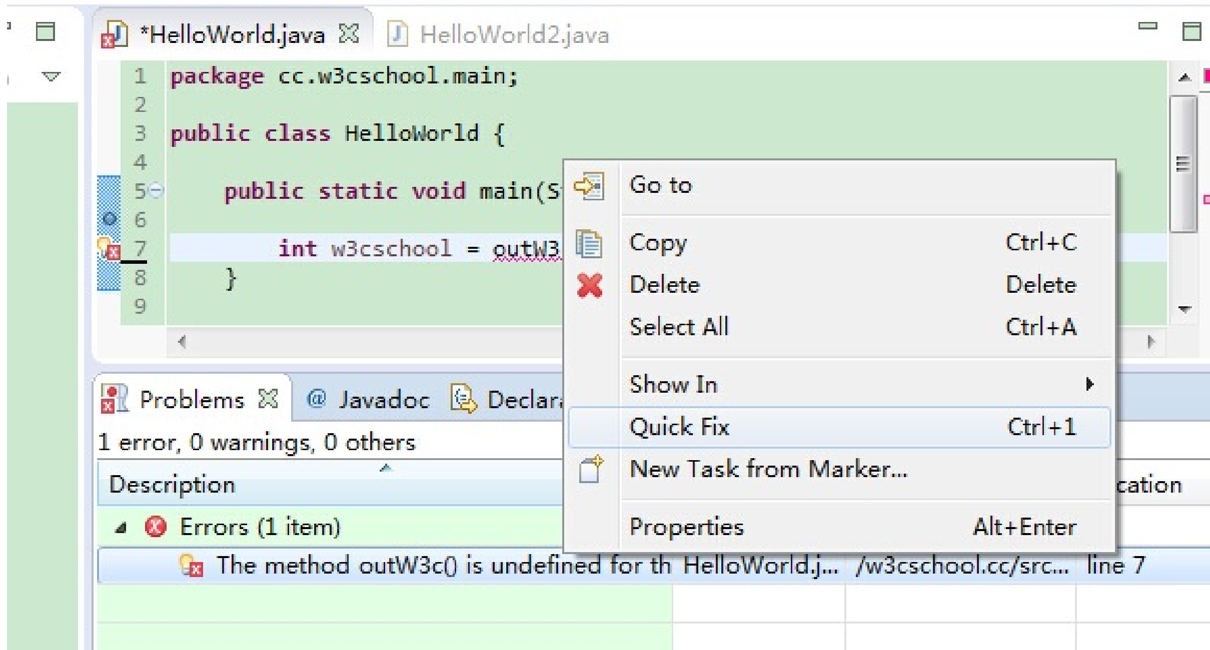
快速修复的对话框提供了解决的方案。快速修复对话框可通过以下方式调用：

- 将鼠标指针放在波浪线上
- 点击小灯泡
- 将鼠标指针放在突出的文本上并选择 Edit 菜单上的 Quick fix 项或者按下快捷键 Ctrl + 1



在上图中，getId 被高亮显示，因为 Person 类中没有一个名为的 getId() 方法。 在弹出的修复方案中选择 "Create method 'getId()' in type 'Person'" 这样就能在 Person 类中添加 getId() 方法。

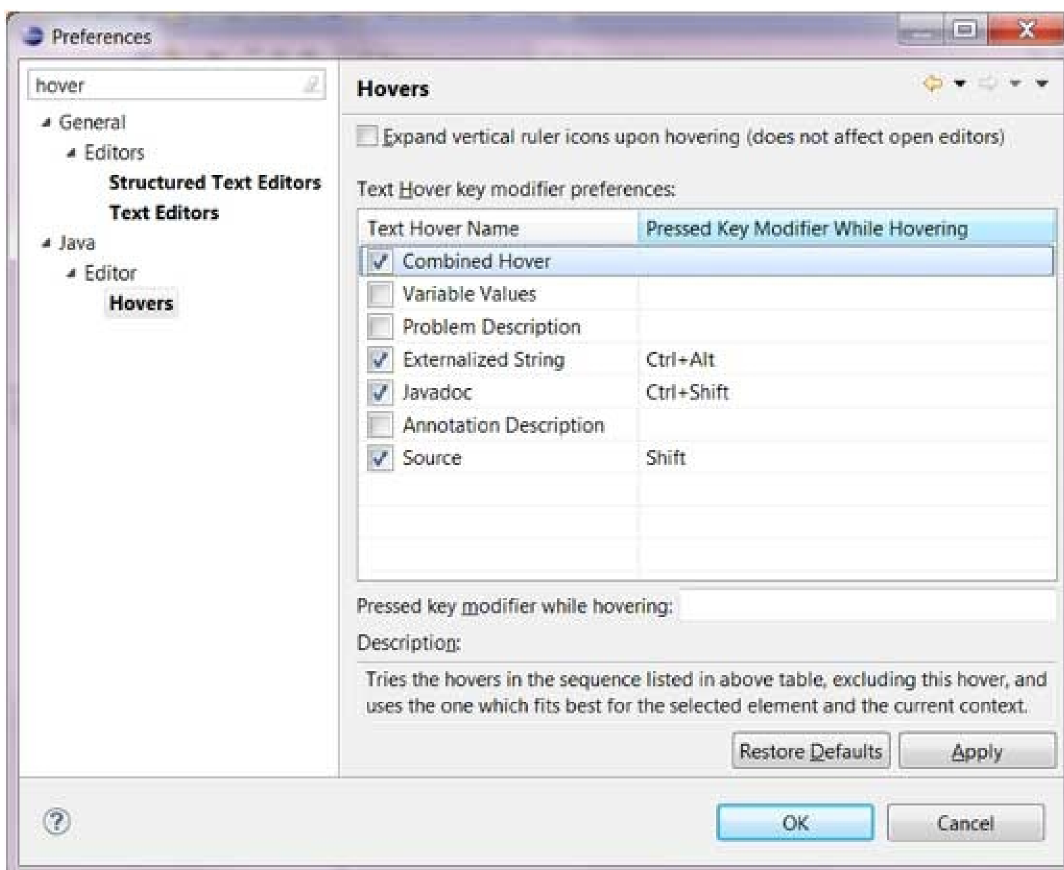
也可以通过右键点击 Problems 视图中的错误项，然后选择快速修复菜单项显示的快速修复对话框，如下图所示：



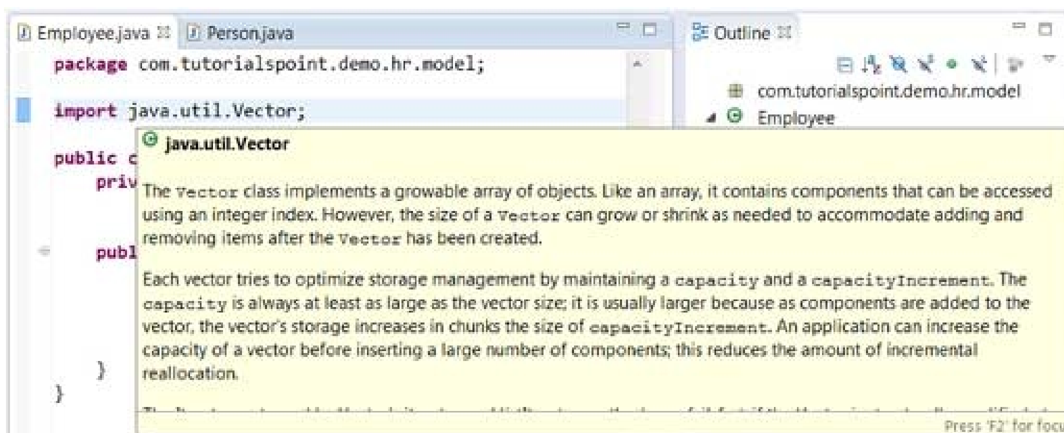
Eclipse 悬浮提示

使用悬浮提示

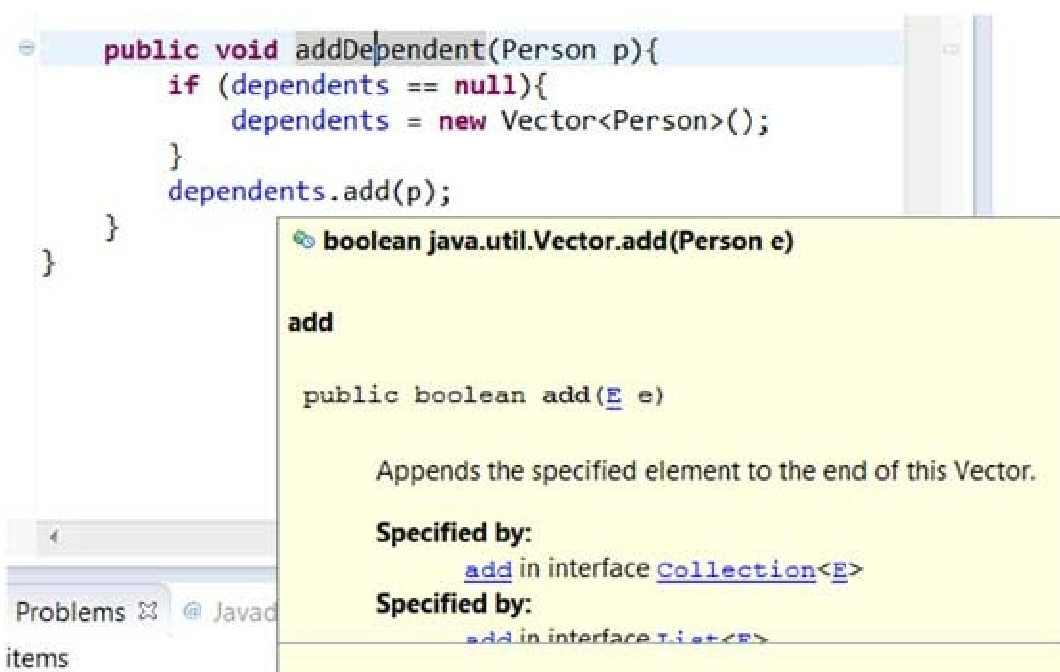
java 编辑器中包含了不同类型的悬浮提示，悬浮提示提供了鼠标指针指向元素的额外信息。所有java编辑器中相关的悬浮提示可以通过 preference(首选项) 的 Hovers 页面来配置（搜索框中输入 "hover"）。



java 编辑器中将鼠标指针移至类上，将显示与该类相关的java文档信息。



java 编辑器中将鼠标指针移至方法上，将显示与该方法相关的java文档信息。


public void addDependent(Person p){
if (dependents == null){
dependents = new Vector<Person>();
}
dependents.add(p);
}
}
boolean java.util.Vector.add(Person e)
add
public boolean add(E e)
Appends the specified element to the end of this Vector.
Specified by:
add in interface Collection<E>
Specified by:
Problems
items

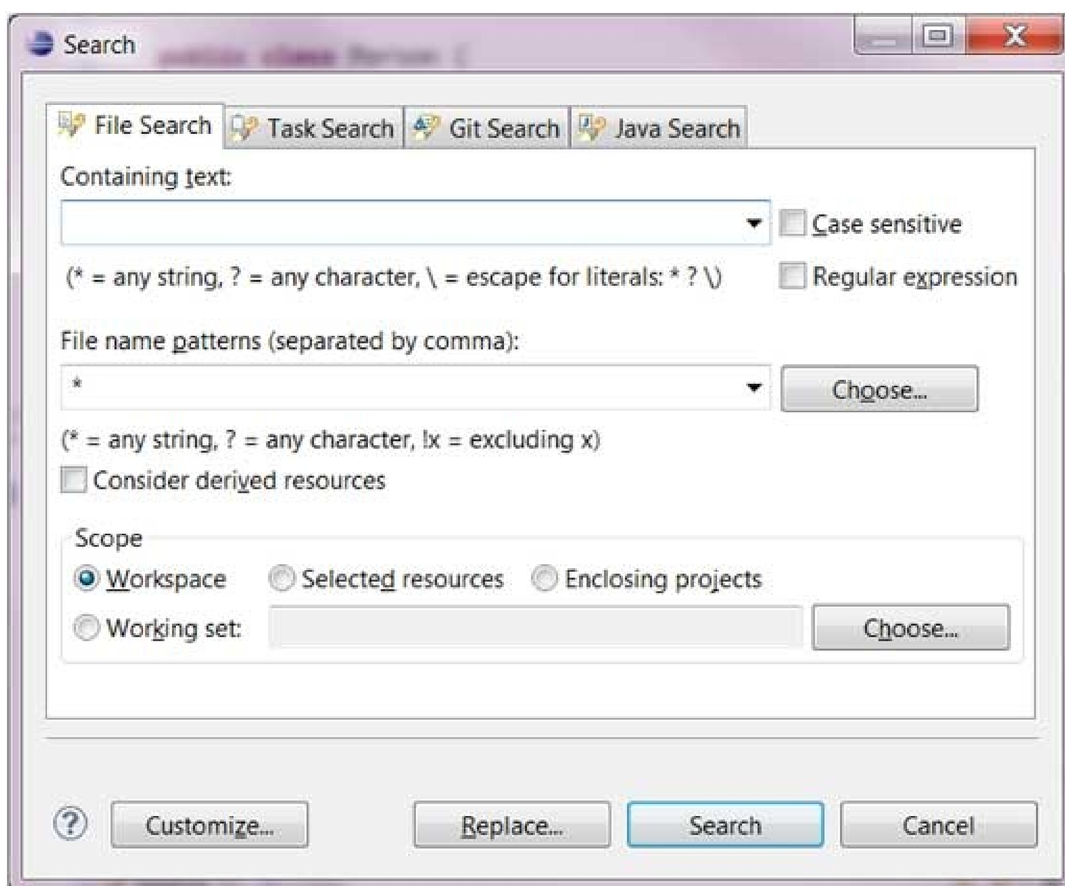
Eclipse 查找

工作空间中查找

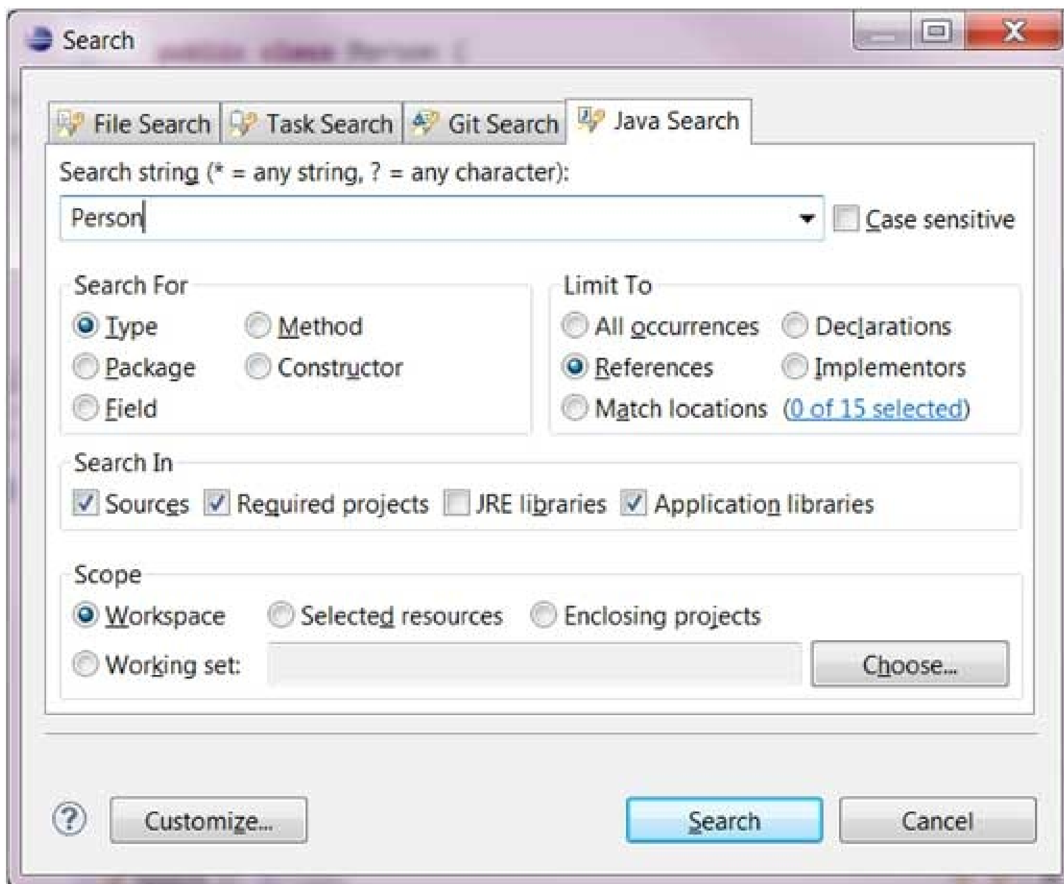
Eclipse 查找对话框中可以允许用户在指定工作空间上使用单词或字母模式来查找文件。 或者你可以在指定项目或在 package explorer 视图上选择好指定文件夹来查找。

可通过以下方式来调用查找框：

- 在 Search 菜单上选择 Search 或 File 或 Java
- 按下快捷键： Ctrl + H



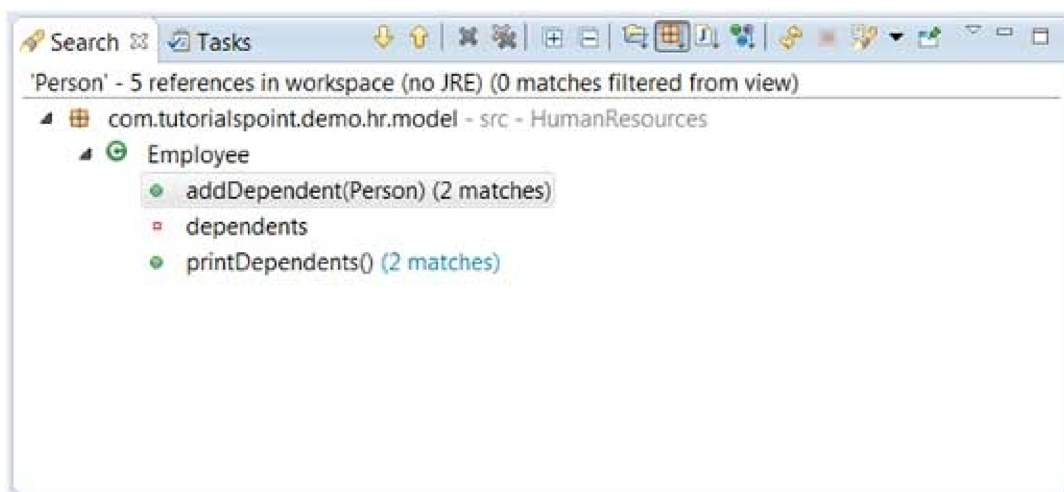
文件(File)查找允许用户查找所有文件类型，而 Java 查找只针对 Java 文件进行查找。



例如我们查找 Person 类型使用的情况，可以通过 Java 查找页面：

- 在查找框中输入 Person
- 在 search for 的单选按钮中选择 Type
- 在 limit to（限于）单选按钮中选择 References
- 点击 Search

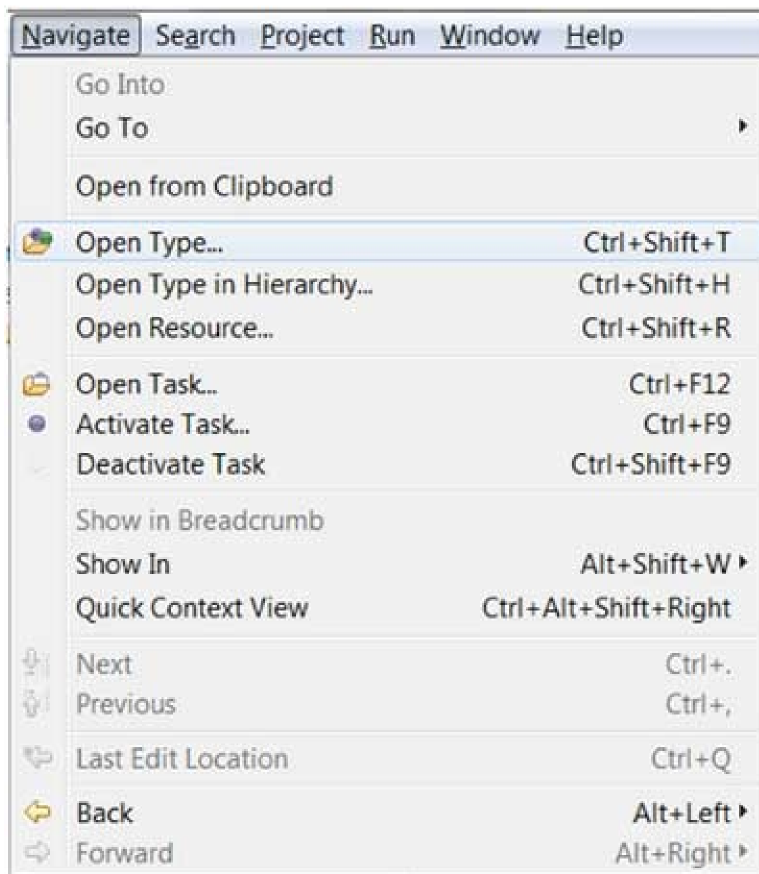
Search 视图中显示结果如下：



Eclipse 浏览(Navigate)菜单

浏览 Eclipse 工作空间

浏览(Navigate)菜单提供了多个菜单可以让你快速定位到指定资源。

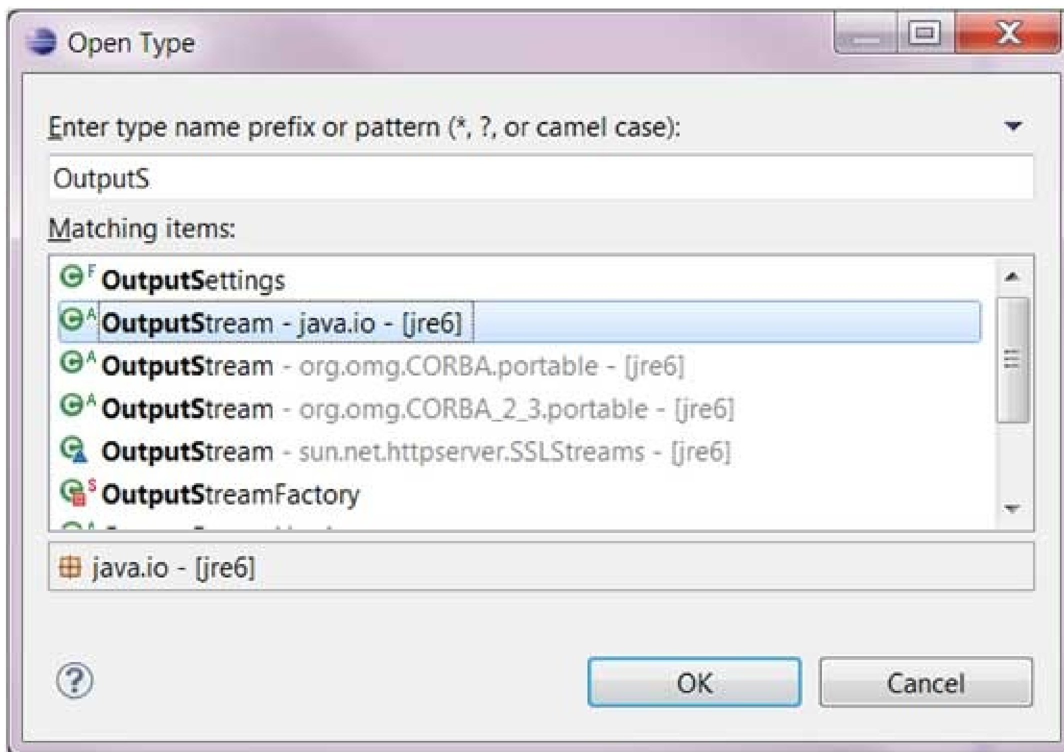


上图中 Open Type, Open Type in Hierarchy 和 Open Resource 三个菜单项是非常有用的。

Open Type

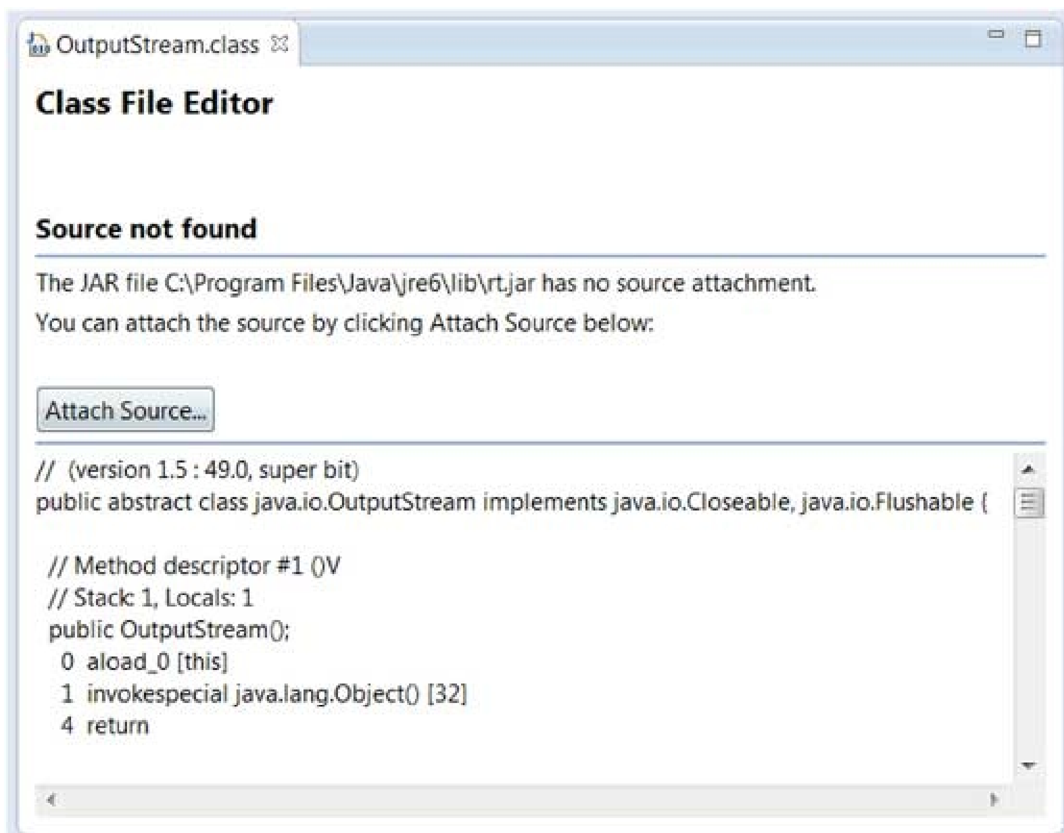
Open Type 菜单项可以打开一个对话框，对话框中可以查找 Java 类型文件。

你可以在输入框中输入类名查找。 '*' 号表示 0 个或多个字母，'?' 号表示单个字母可用于指定模式。对话框中将显示所有匹配的模式。



你列表中选择你查找的文件即可。

Eclipse 将打开一个编辑器，显示所选择的类型。如果所选类型不能显示源代码，将使用类文件编辑器显示所选类型的字节码。

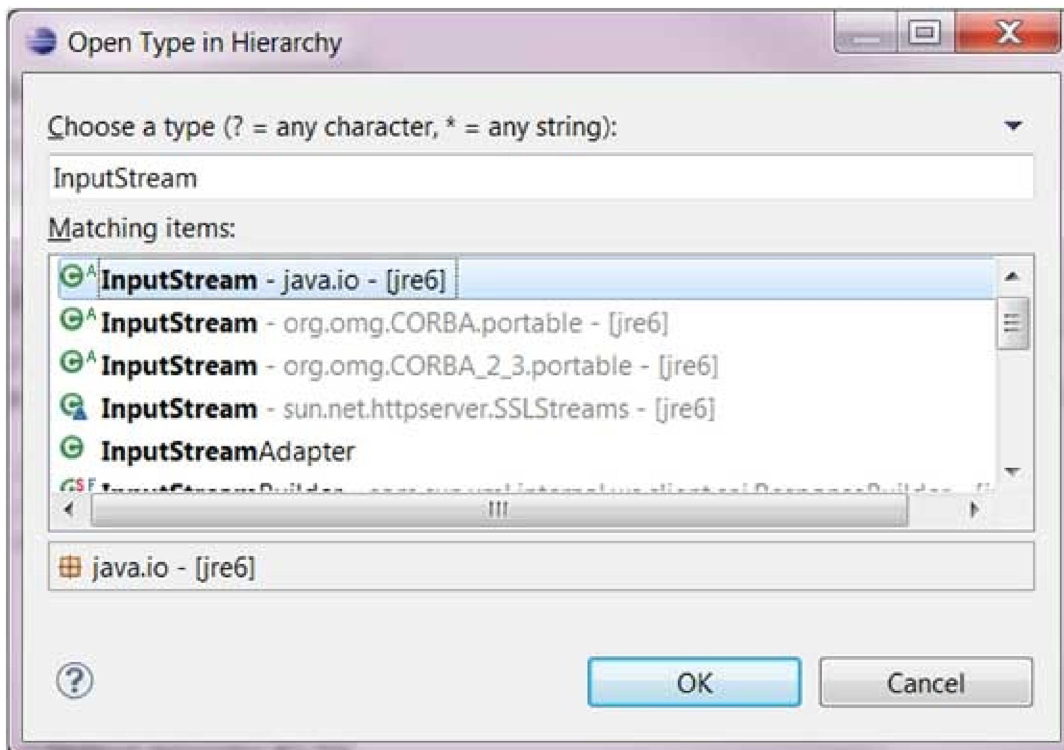


你可以点击 Attach Source 按钮来查看类文件对应的源码。

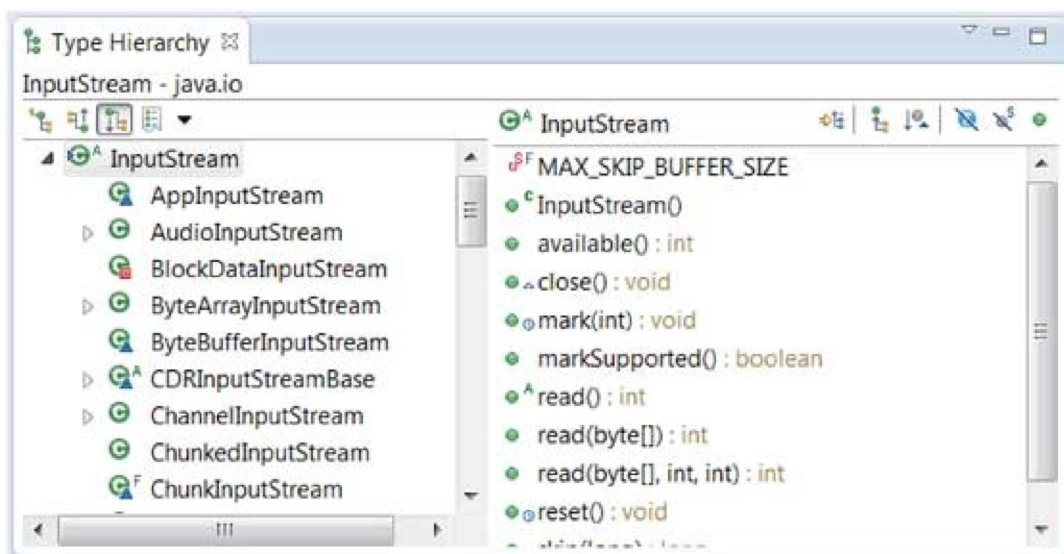
源代码位于 Java 主目录中的 src.zip 压缩文件中。

Open Type in Hierarchy

Open Type in Hierarchy 菜单允许用户在 Type Hierarchy 视图中查看类的继承层次。



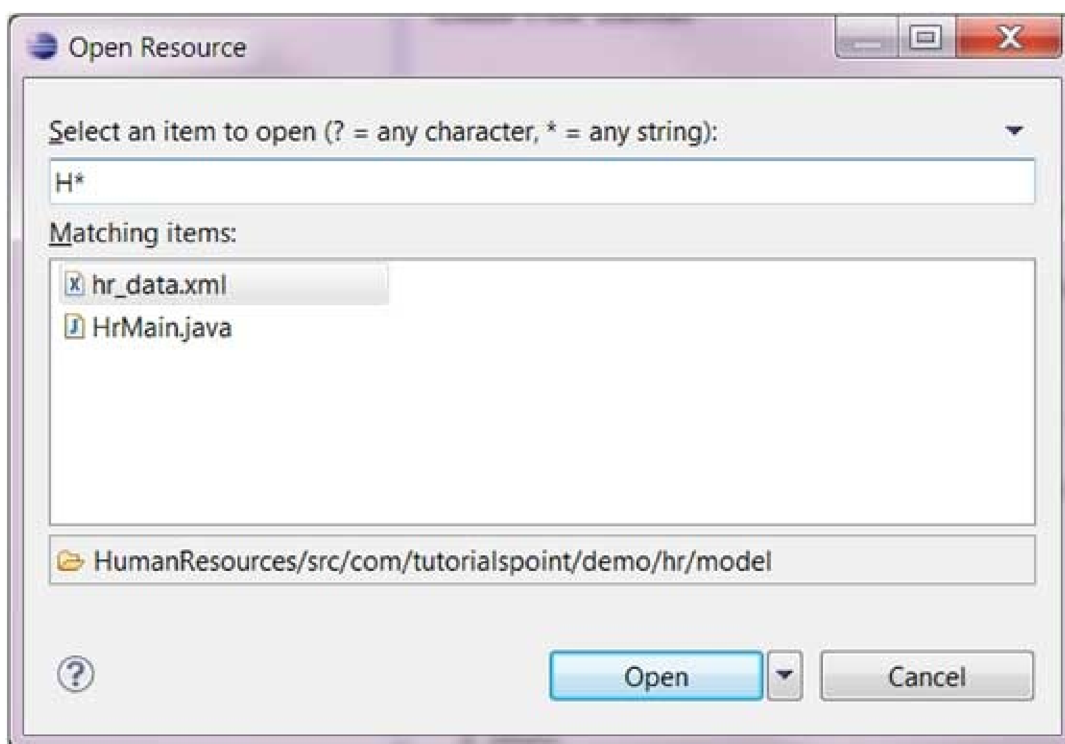
Type Hierarchy视图中选择指定的类就可以看到类的定义信息，包含对应的属性和方法：



Open Resource

open resource(打开资源)菜单可用于查找工作空间中的文件。

'*' 号表示 0 个或多个字母，'?' 号表示单个字母可用于指定模式。对话框中将显示所有匹配的模式。



选择你要打开的文件并点击 OK 按钮。

Eclipse 重构菜单

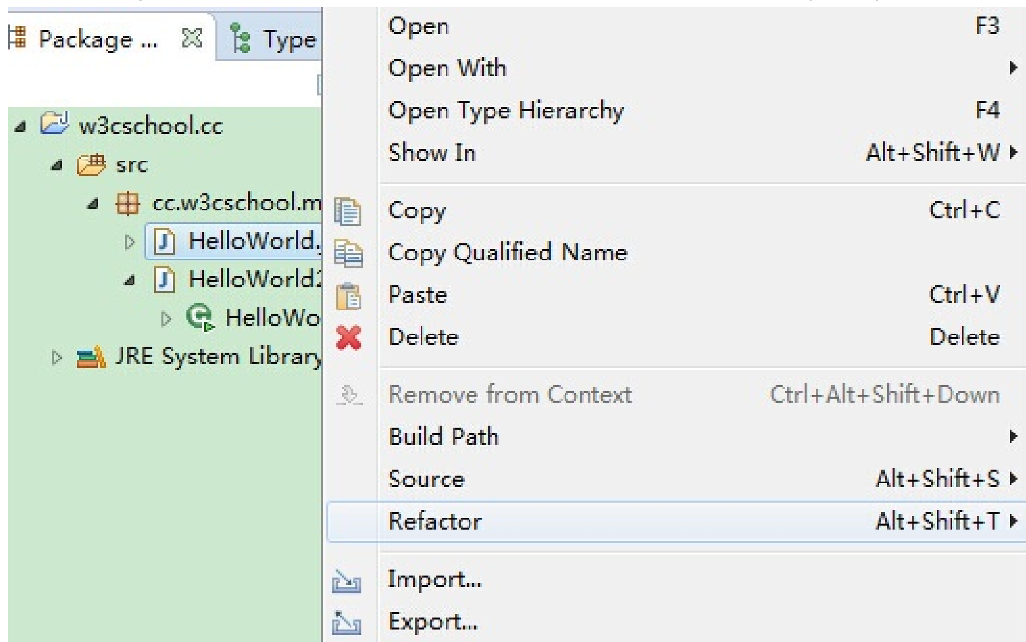
使用Eclipse重构

在项目开发中我们经常需要修改类名，但如果其他类依赖该类时，我们就需要花很多时间去修改类名。

但 Eclipse 重构功能可以自动检测类的依赖关系并修改类名，帮我们节省了很多时间。

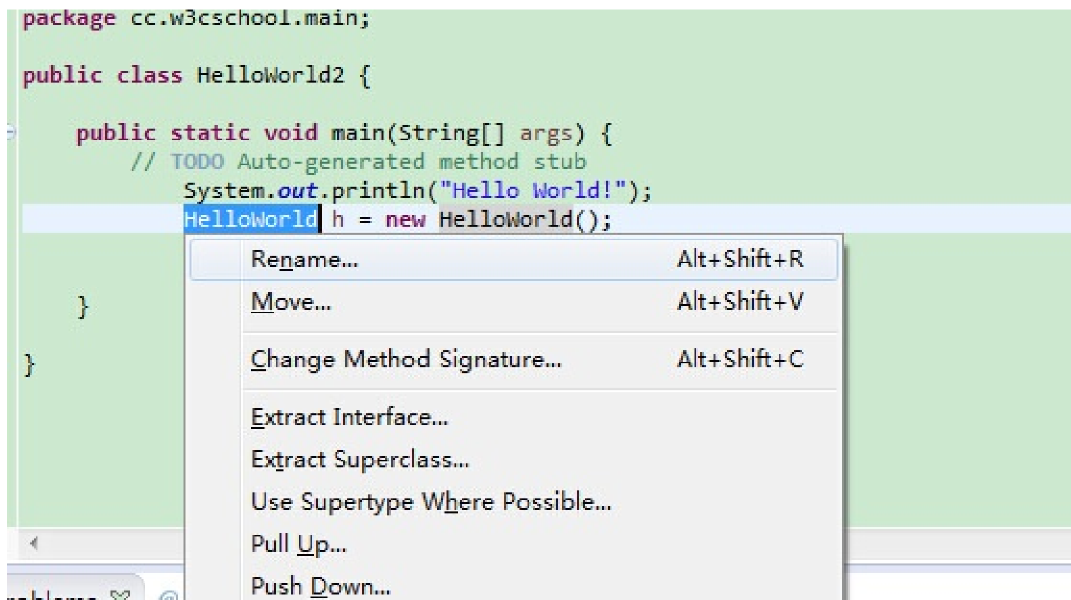
可用过以下方式打开重构菜单：

- 在 Package Explorer 视图中右击 Java 元素并选择Refactor(重构)菜单项

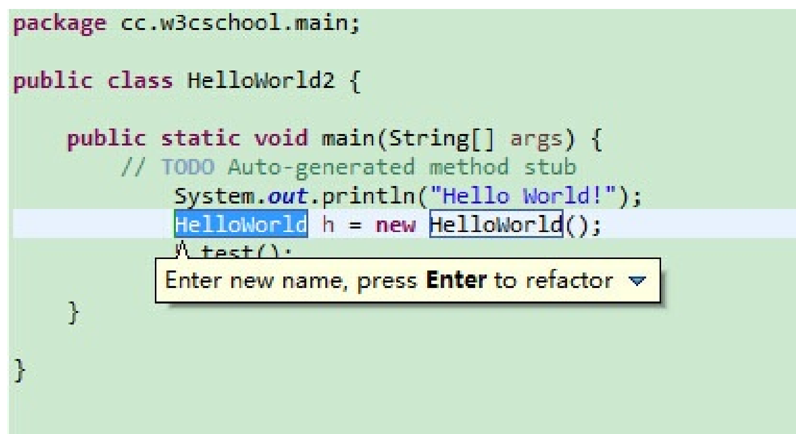


- 在 Java 编辑器中鼠标右击 Java 元素并选择Refactor(重构)菜单项
- 在 Package Explorer 视图中选择 Java 元素并按下 Shift + Alt + T

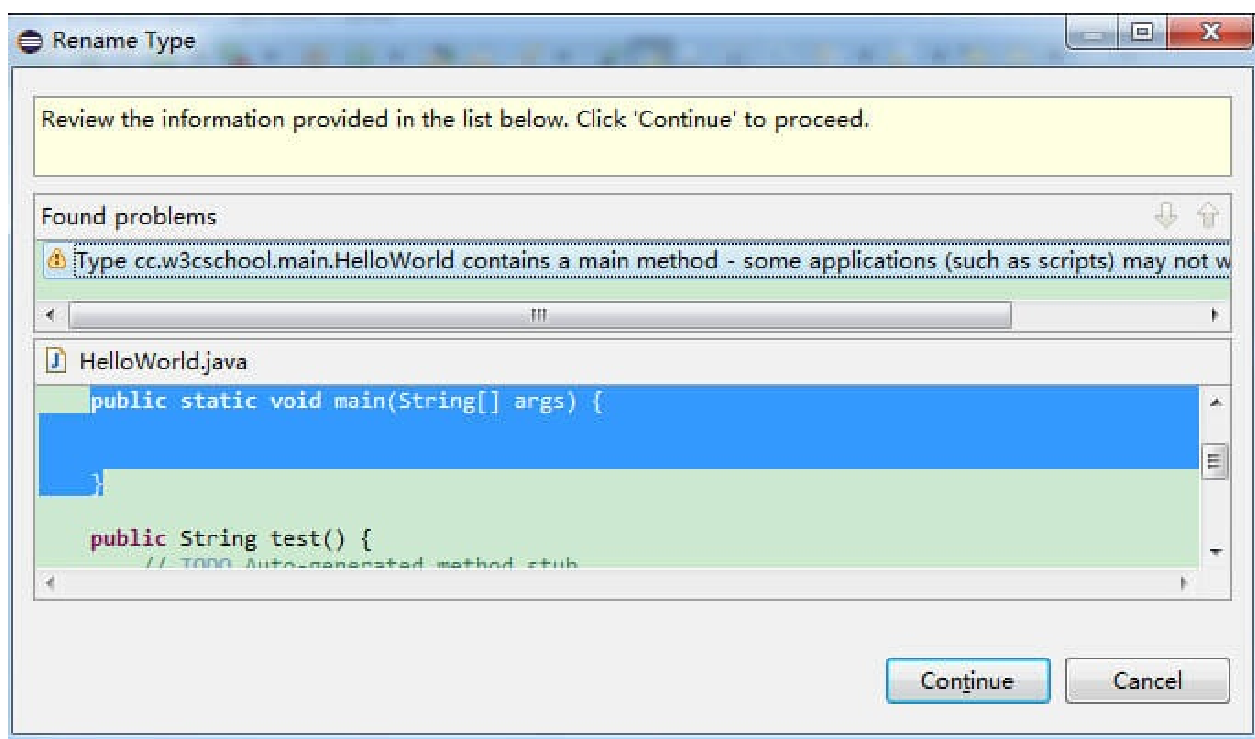
下图中我们在Java 编辑器中选中了 HelloWorld 类：



在选择 Rename 后会提示输入新的类名并按回车结束修改：



在修改完成按下回车键后，会弹出将将会修改的类：



你只需点击 Continue 按钮即可完成操作。

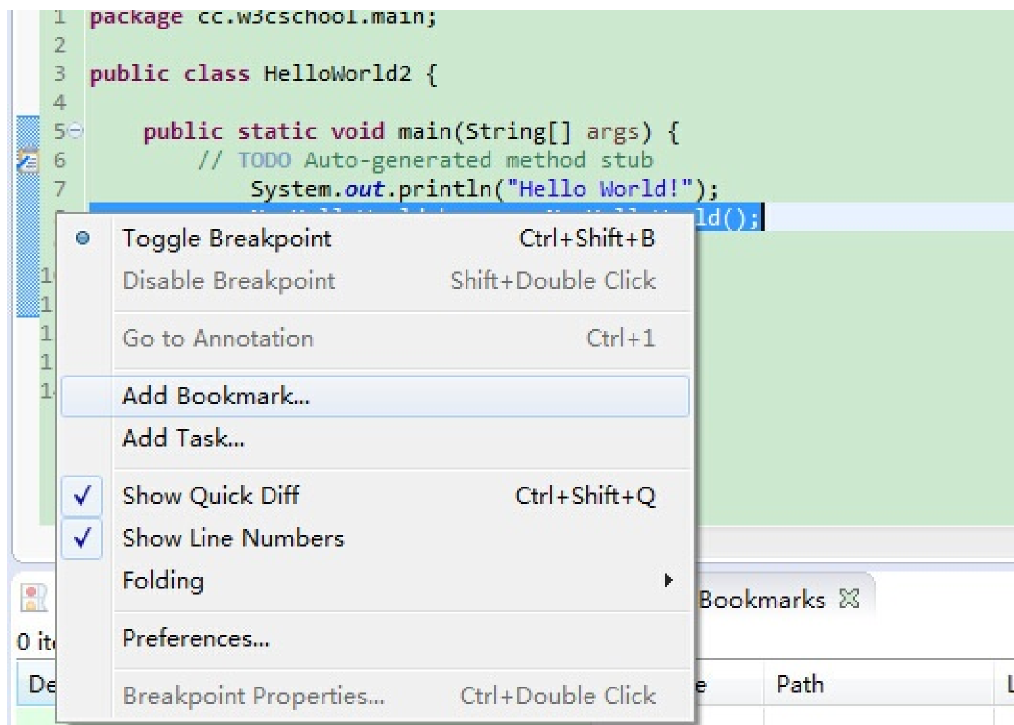
Eclipse 添加书签

关于书签

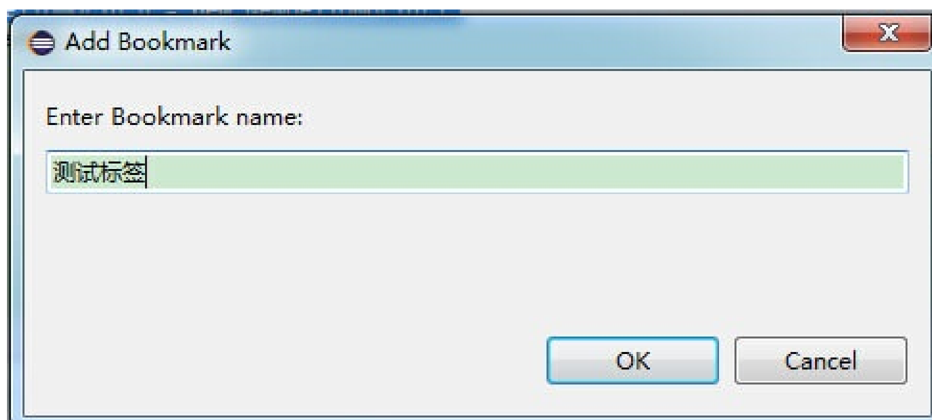
Eclipse 中可以在编辑器的任意一行添加书签。 您可以使用书签作为提示信息，或者使用书签快速定位到文件中的指定的行。

添加书签

如果你想设置书签，你只需要在垂直标尺上右击鼠标并选择能 "Add Bookmark" 即可。



在弹出的对话框中输入书签名。



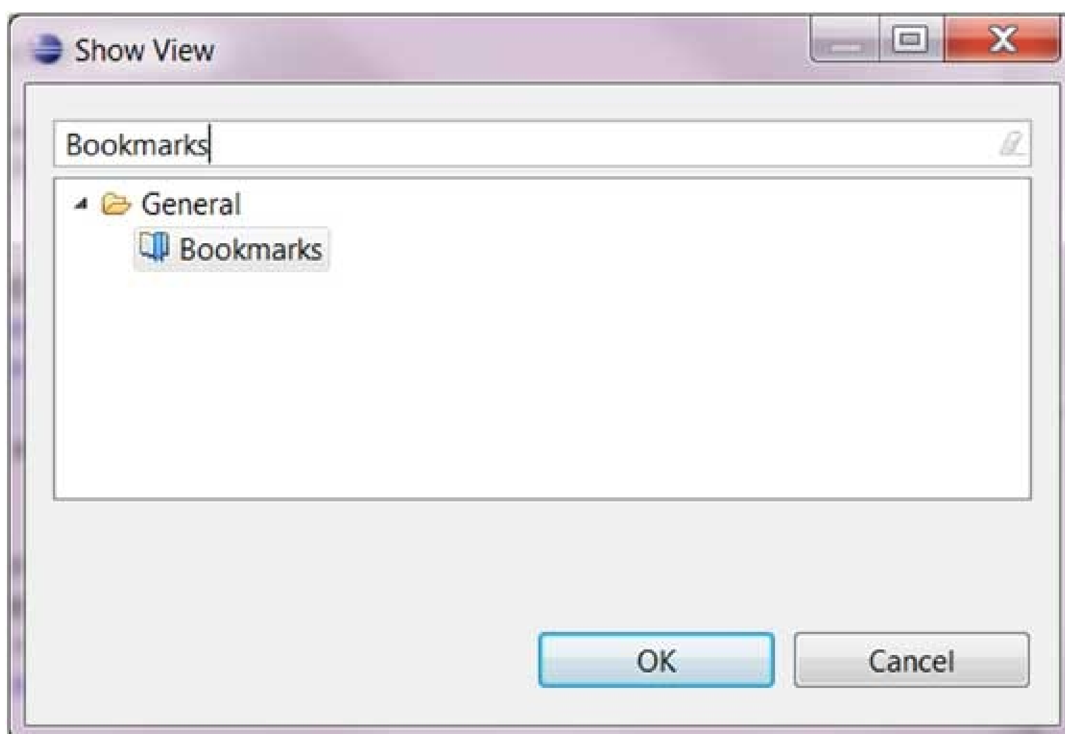
垂直标尺上就会出现一个书签的按钮，当然你也可以在 Bookmarks 视图中查看到书签列表。

```
1  package cc.w3cschool.main;
2
3  public class HelloWorld2 {
4
5      public static void main(String[] args) {
6          // TODO Auto-generated method stub
7              System.out.println("Hello World!");
8  测试标签      NewHelloWorld h = new NewHelloWorld();
9              h.test();
10
11     }
12
13 }
14
```

打开 Bookmarks(书签) 视图

打开 Bookmarks 视图的方法为:

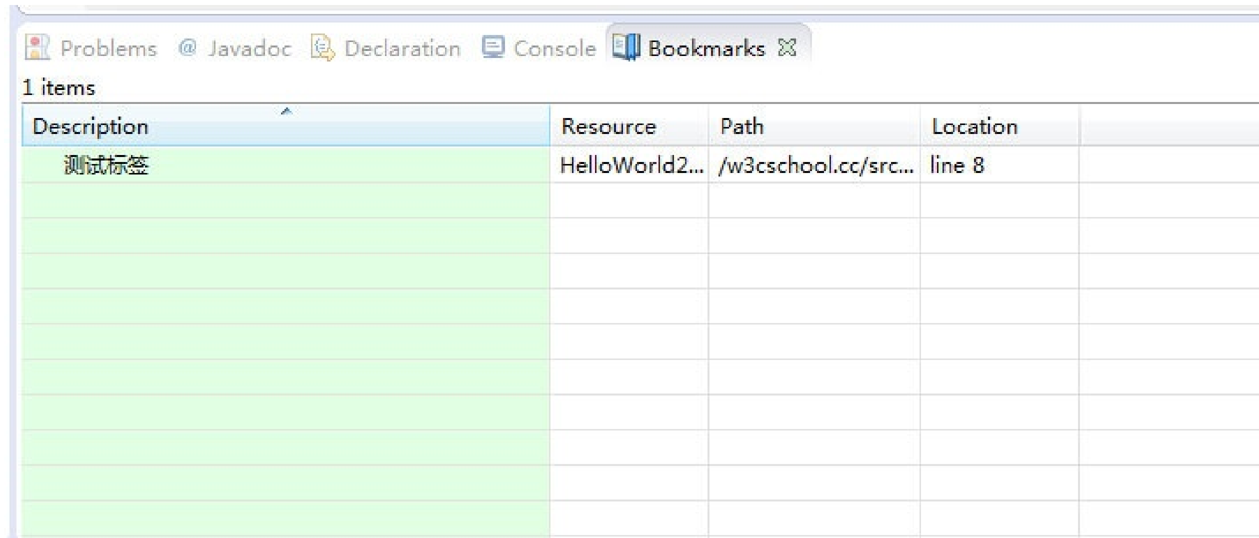
- 点击 Window 菜单选择 Show View > Other
- 在搜索输入框中输入 Bookmark
- 在 General 下选择 Bookmarks



- 点击 OK 按钮

使用 Bookmarks(书签) 视图

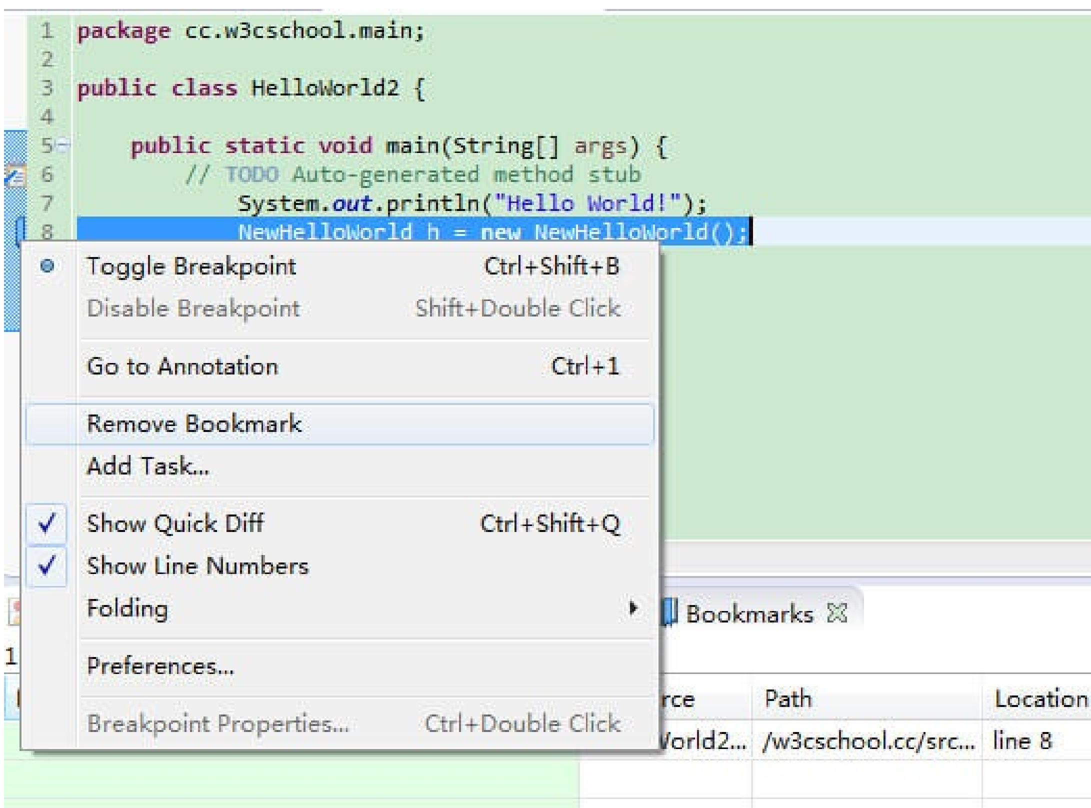
Bookmarks 视图如下：



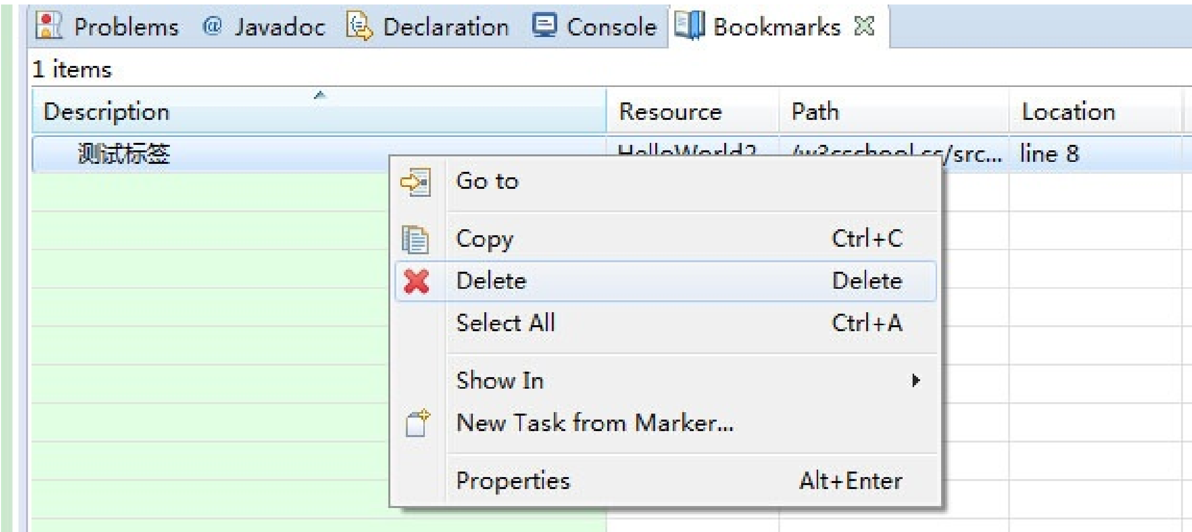
你可以在 Bookmarks 视图中双击书签或者鼠标右击书签选择"Go to"菜单来快速定位书签所在的位置。

删除Bookmarks(书签)

你可以在垂直书签上右击编辑并选择 Remove Bookmark 来删除书签：



或者在 Bookmarks 视图视图中右击书签并选择"Delete"菜单项来删除书签：

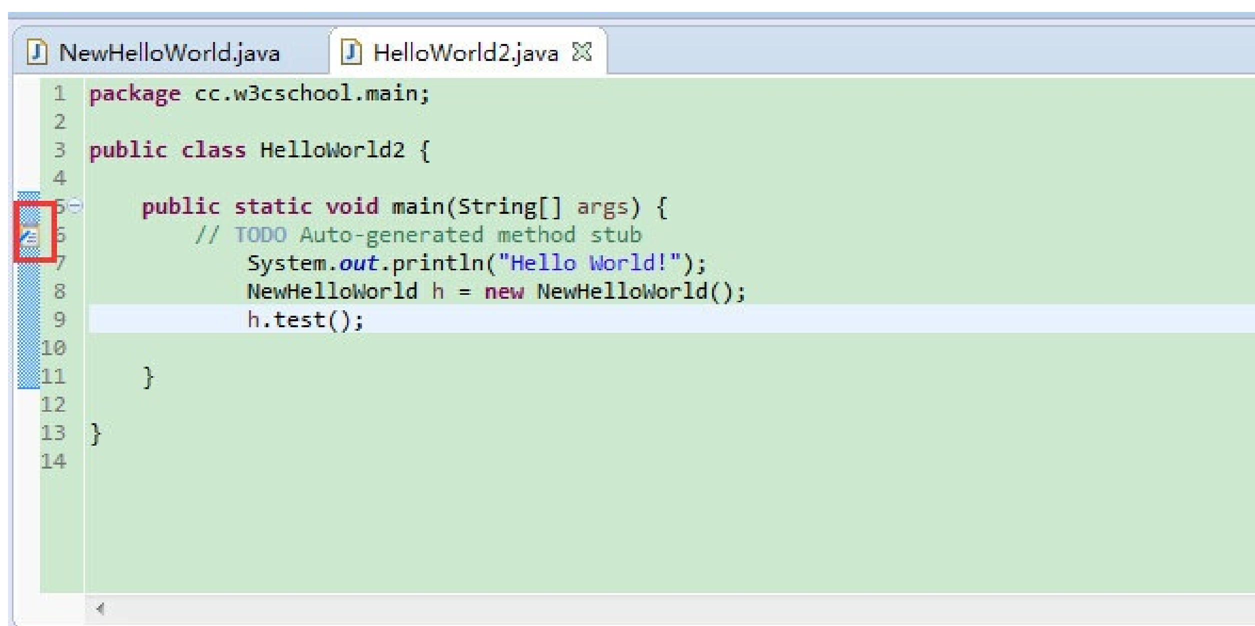

Problems
Javadoc
Declaration
Console
Bookmarks
1 items
Description
Resource
Path
Location
测试标签
/w3cschool.cc/src...
line 8
Go to
Copy Ctrl+C
Delete Delete
Select All Ctrl+A
Show In
New Task from Marker...
Properties Alt+Enter

Eclipse 任务管理

管理任务

在Eclipse中用TODO标签管理任务，利用这个功能可以方便地将项目中一些需要处理的任务记录下来。

我们可以在 Java 代码中的注释添加 TODO 单词来标记一个任务，任务可以通过 Tasks(任务)视图查看。



在Eclipse中我们可以通过鼠标右击垂直标尺并选择 Add Task 菜单来添加任务，在弹出的对话框中输入任务描述信息：

NewHelloWorld.java HelloWorld2.java

```
1 package cc.w3cschool.main;
2
3 public class HelloWorld2 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello World!");
8         NewHelloWorld h = new NewHelloWorld();
```

Toggle Breakpoint Ctrl+Shift+B
Disable Breakpoint Shift+Double Click
Go to Annotation Ctrl+1
Add Bookmark...
Add Task...
Show Quick Diff Ctrl+Shift+Q

Properties

Description: 测试函数还未完成

Priority: Normal Completed

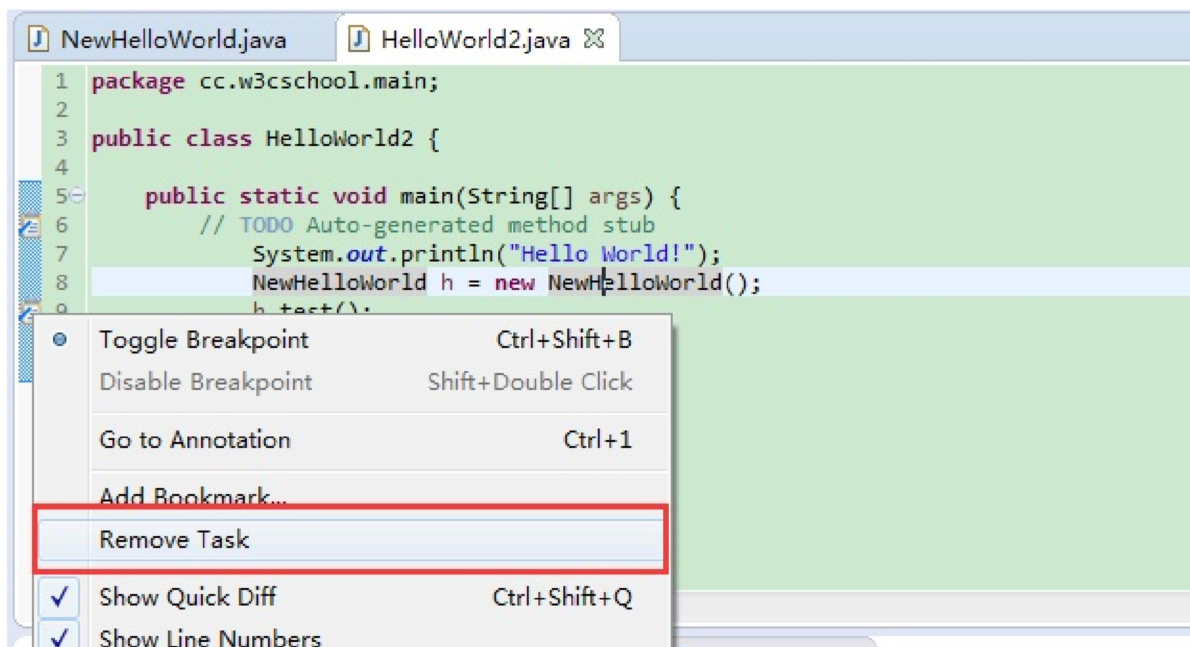
On element: HelloWorld2.java

In folder: w3cschool.cc/src/cc/w3cschool/main

Location: line 9

OK Cancel

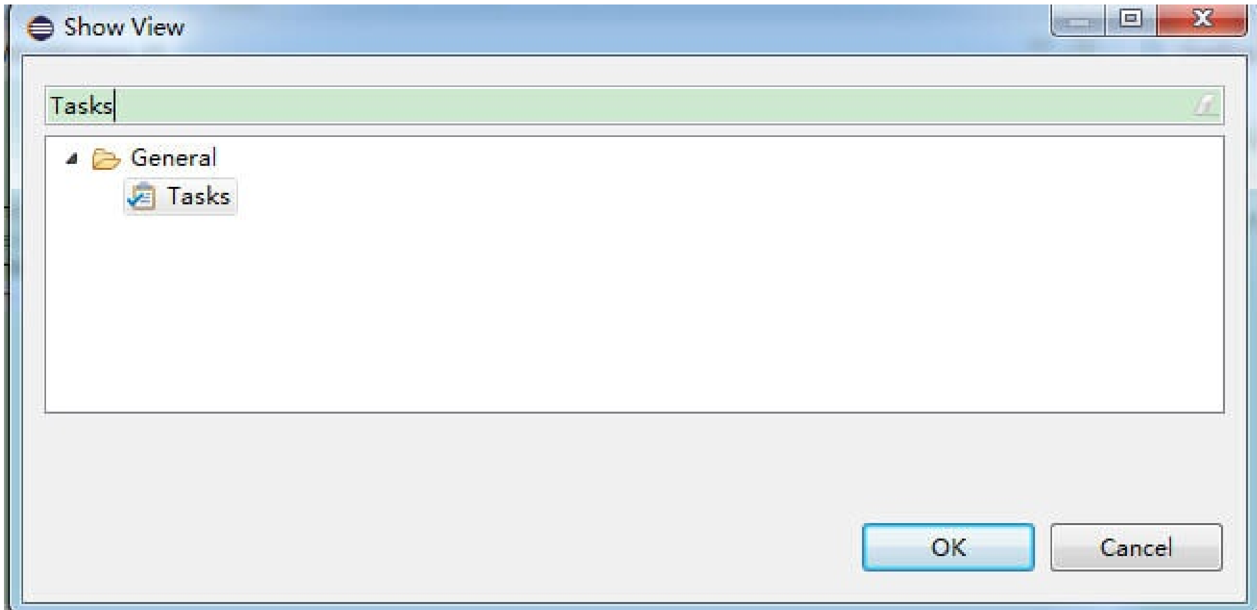
如果需要删除任务，只需右击任务图标选择 Remove Task 菜单项即可：



打开 Task(任务) 视图

打开 Task(任务) 视图的方法为：

- 在 Window 菜单中选择 Show View > Other
- 在搜索框中输入 Tasks
- 在 General 下选择 Tasks



- 最后点击 OK 按钮

使用 Task(任务) 视图

Task(任务) 视图中显示了项目中所有待完成的任务：

Problems | @ Javadoc | Declaration | Console | Tasks

3 items

	!	Description	Resource	Path	Location	Type
☐		测试函数还未完成	HelloWorld2...	/w3cschool.cc/src...	line 9	Task
		TODO Auto-generated method stub	HelloWorld2...	/w3cschool.cc/src...	line 6	Java Task
		TODO Auto-generated method stub	NewHelloW...	/w3cschool.cc/src...	line 11	Java Task

Task(任务) 视图中还能进行以下操作：

- 修改任务右下角
- 标记任务已完成
- 删除任务或删除所有已完成任务

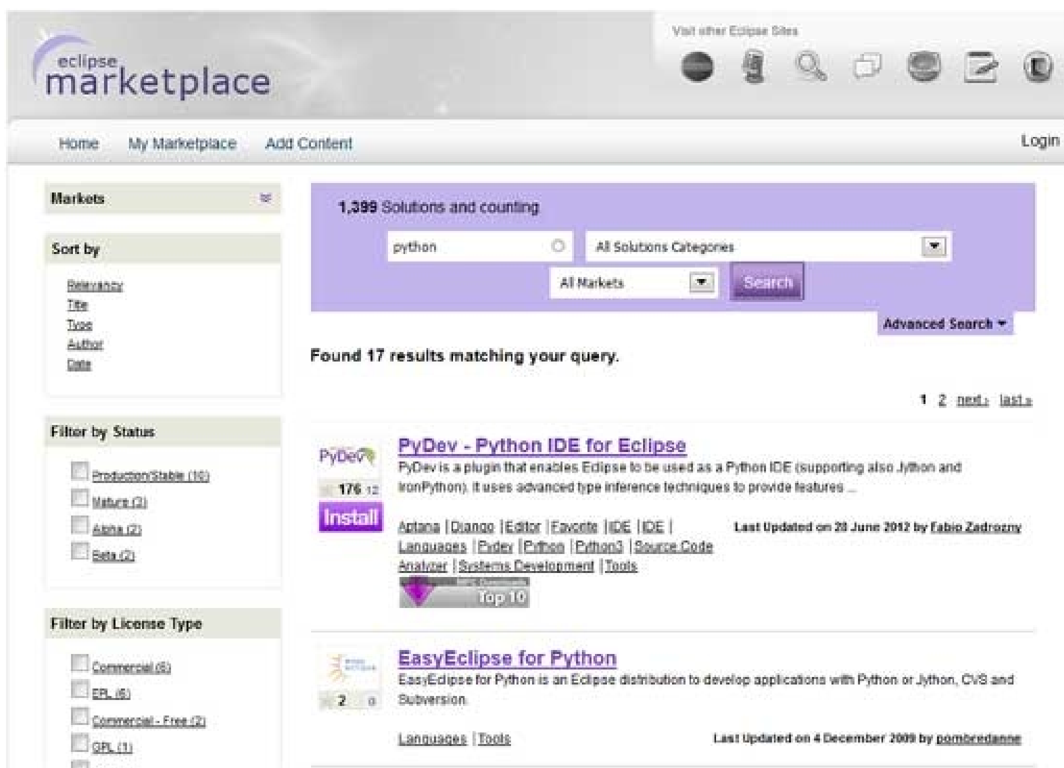
Eclipse 安装插件

查找和安装插件

Eclipse作为一个集成的IDE开发工具，为我们的软件开发提供了便利，eclipse除了自带的强大功能外，还支持功能丰富的插件。

我们可以通过Eclipse官方市场 (http://marketplace.eclipse.org/)找到并下载我们需要的插件。

例如我们可以查找支持 Python IDE 的插件，如下图所示：



在 Eclipse IDE 中我们也可以通过点击 Help 菜单中的 Eclipse Marketplace（Eclipse 超市）选项来查找插件：

Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Finish to proceed with installation.
Press the information button to see a detailed overview and a link to more information.

Search | Recent | Popular | Installed

Find: python   All Markets   All Categories   Go

PyDev - Python IDE for Eclipse   Share

PyDev is a plugin that enables Eclipse to be used as a Python IDE (supporting also Jython and IronPython). It uses advanced type inference techniques to provide...

by Appcelerator, EPL   Install

IDE Python Aptana Pydev Django

EasyEclipse for Python   Share

EasyEclipse for Python is an Eclipse distribution to develop applications with Python or Jython, CVS and Subversion.

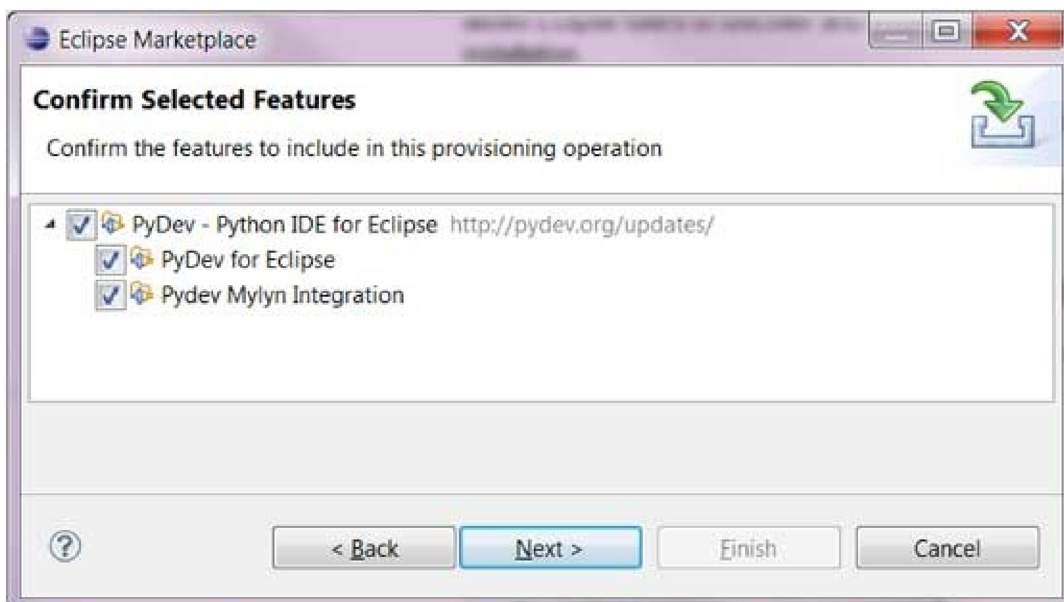
by nexB Inc., Commercial   Learn more

Enerate   Share

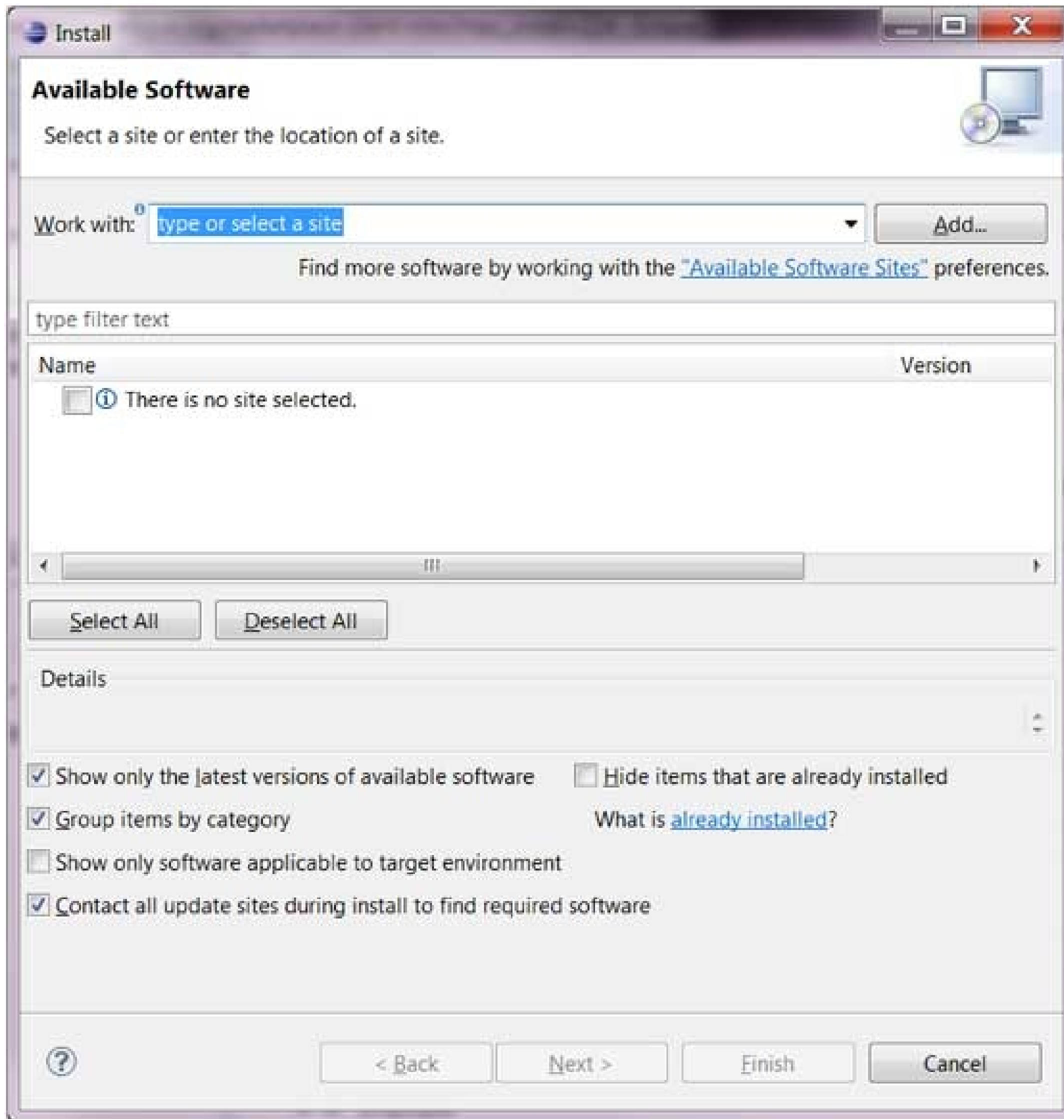
JET2 transforms for generating: - Python code - Python code with Traits - Google

< Back   Next >   Finish   Cancel

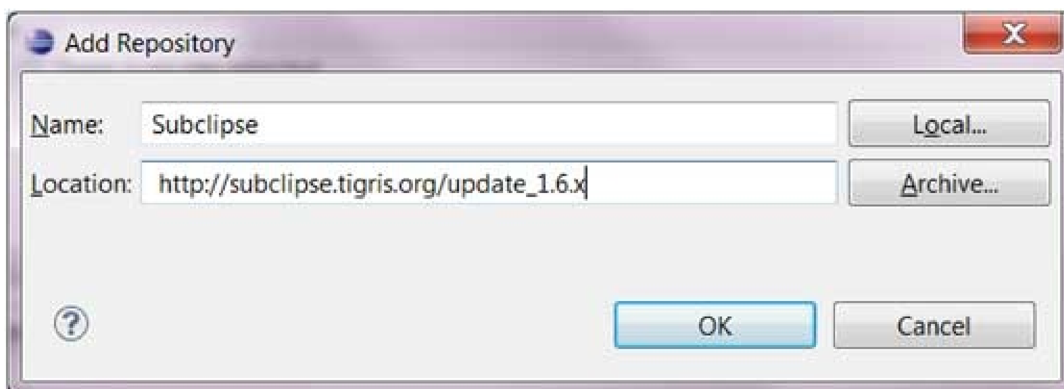
上图中我们选择 PyDev 让 Eclipse 支持 Python 开发，我们只需要点击 Install 按钮即可。 以下对话框为选择安装的插件。



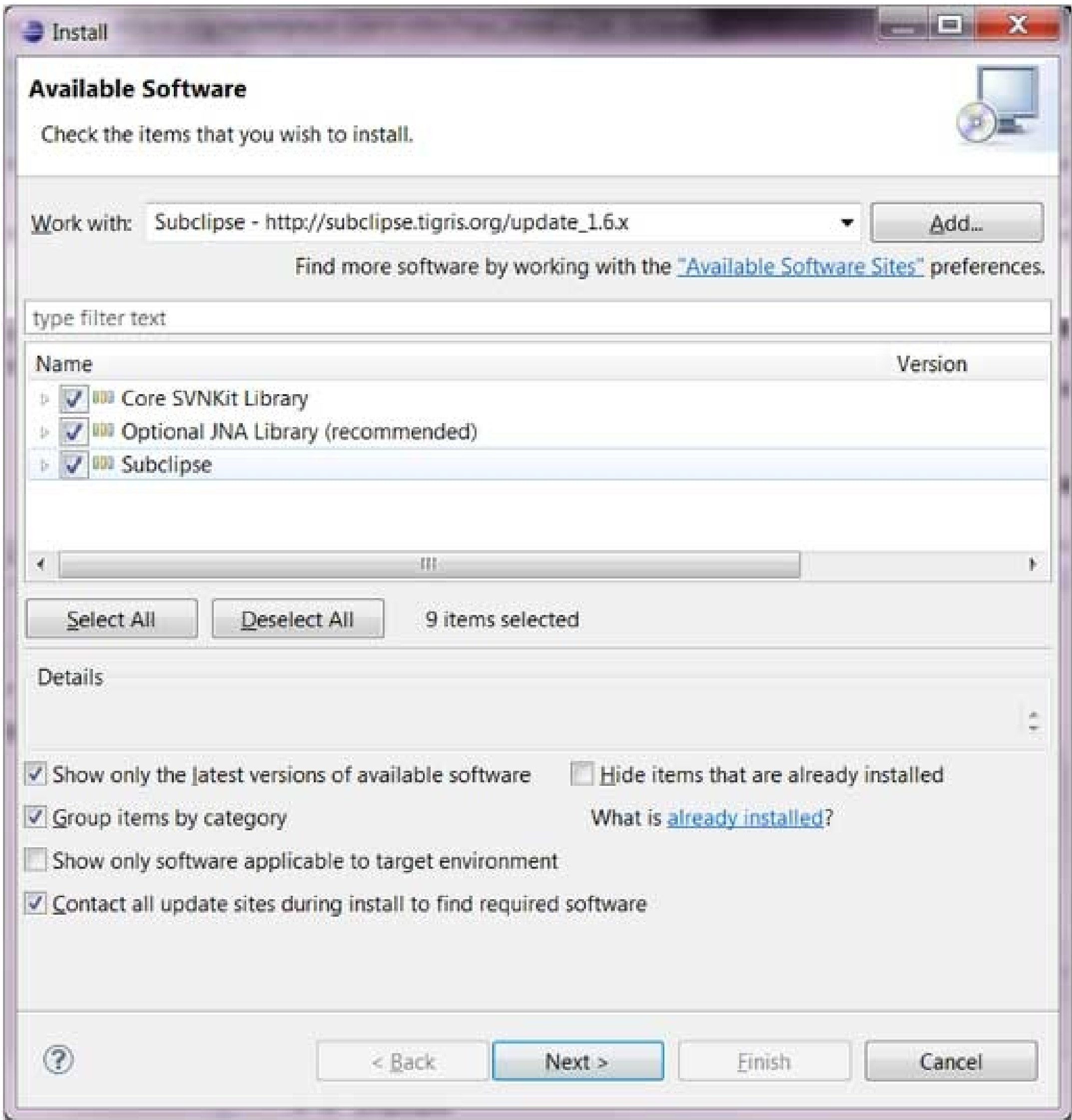
你也可以通过点击 Help 菜单上的 Install New Software 菜单项来安装插件：



这种方式我们需要知道插件远程的安装地址，你可以通过点击 Add 按钮来提交 URL。



安装的对话框中列出了远程可安装的插件列表：


Install
Available Software
Check the items that you wish to install.
Work with: Subclipse - http://subclipse.tigris.org/update_1.6.x
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version
Core SVNKit Library
Optional JNA Library (recommended)
Subclipse
Select All
Deselect All
9 items selected
Details
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Show only software applicable to target environment
Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel

Eclipse 代码模板

使用代码模板

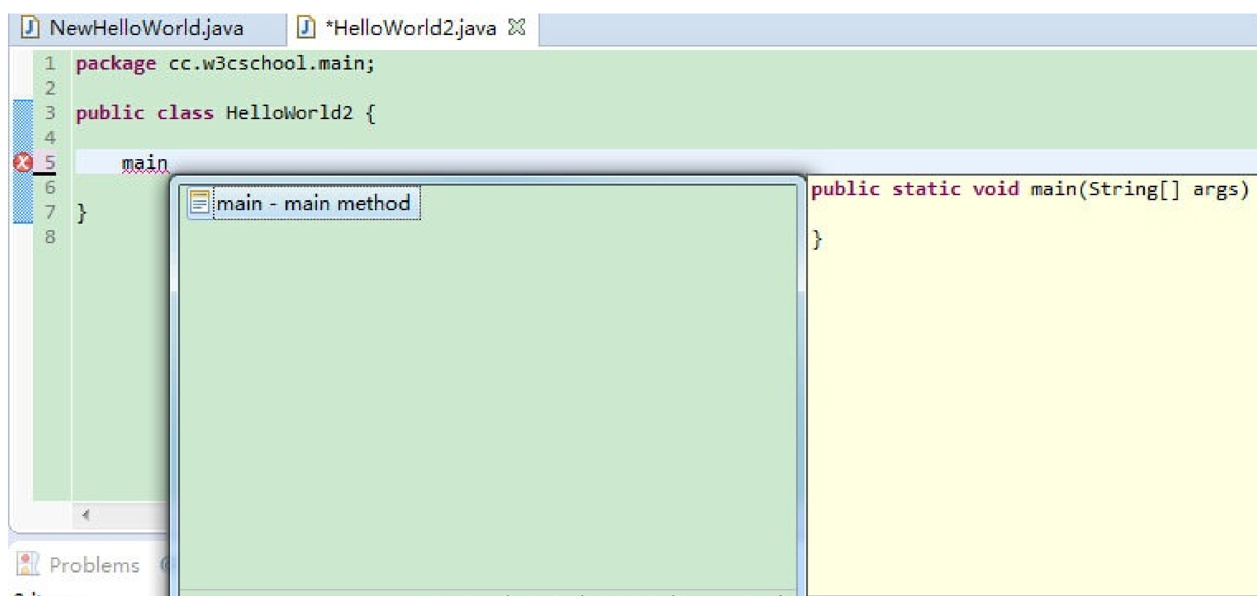
Eclipse 提供了通过定义和使用代码模板来提高工作效率与代码可预测性的能力。

我们在开发 Java 程序过程中经常需要编写 main 方法：

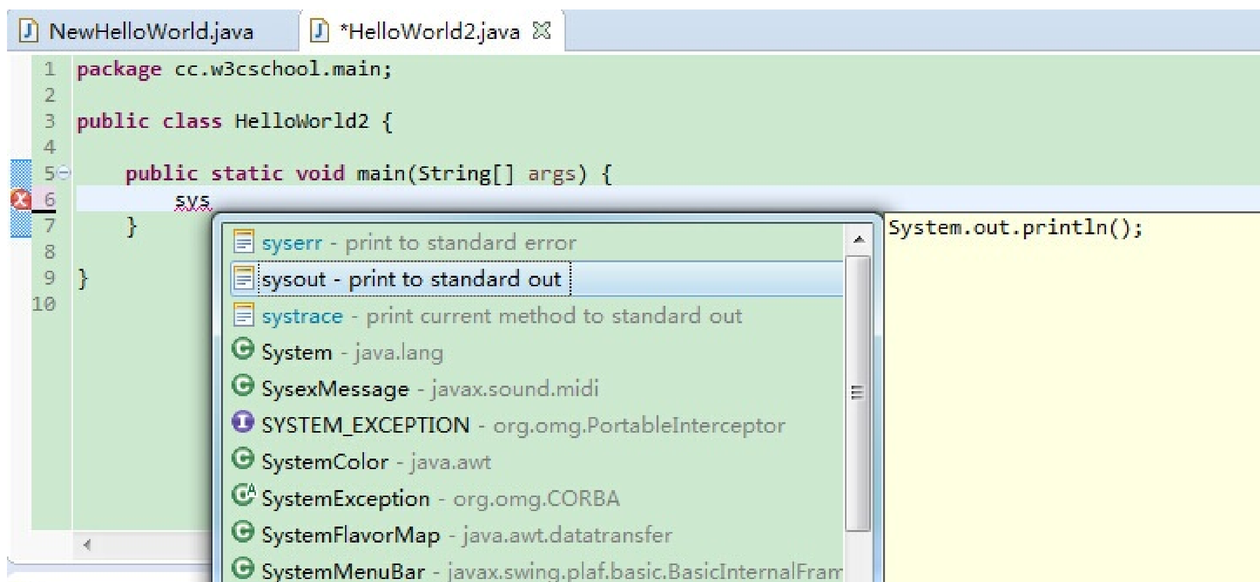
```
public static void main(String[]args) {
}
```

如果我们一个字母一个字母去编写，将是一个重复而又毫无意义的事情，这是我们就可以使用 Eclipse 代码模板来快速完成这些工作。

我们只需在类体中键入main，然后使用Eclipse的代码提示快捷键(默认为Alt+/)，回车后，就可以看到Eclipse自动帮我们完成了main函数的完整定义：

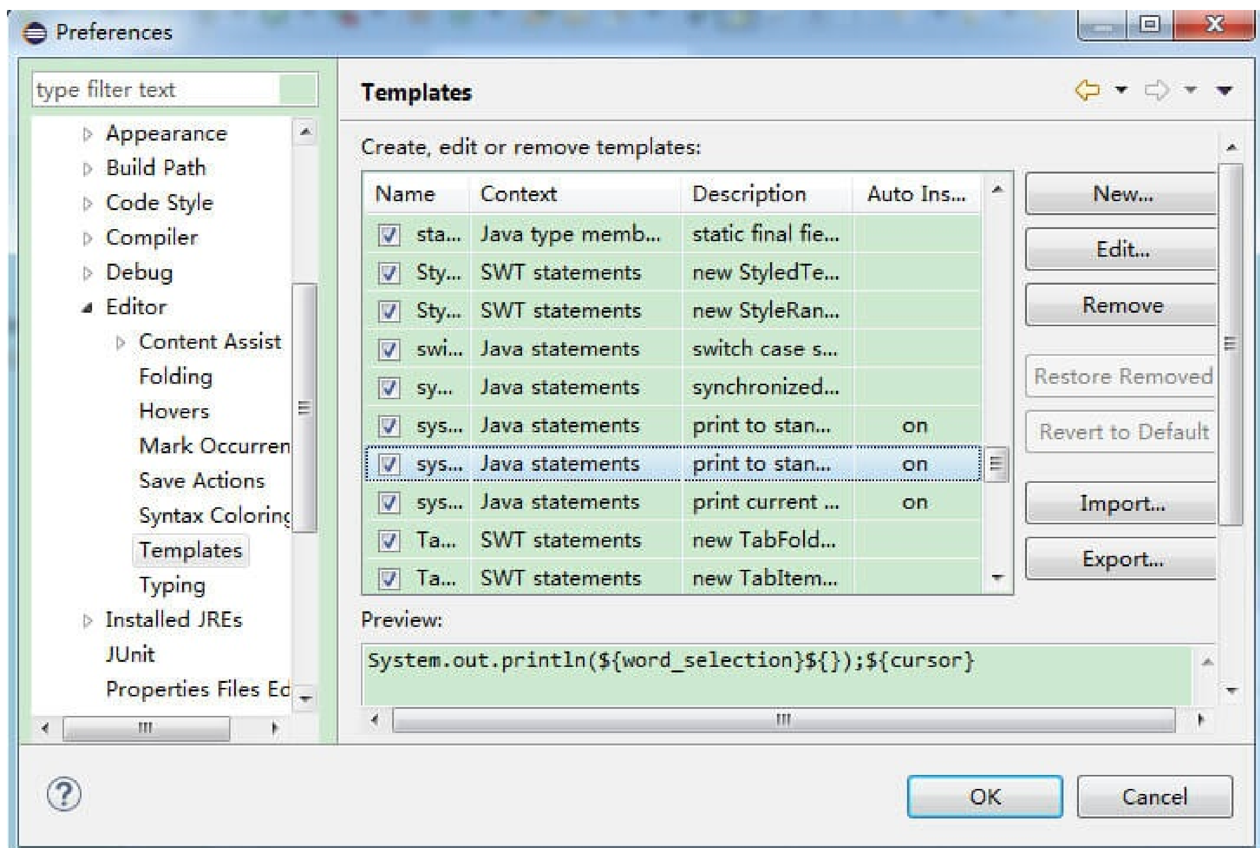


如果我们要使用 System.out.println()，我们只需要输入 syso 然后按下 Alt+/ 即可：

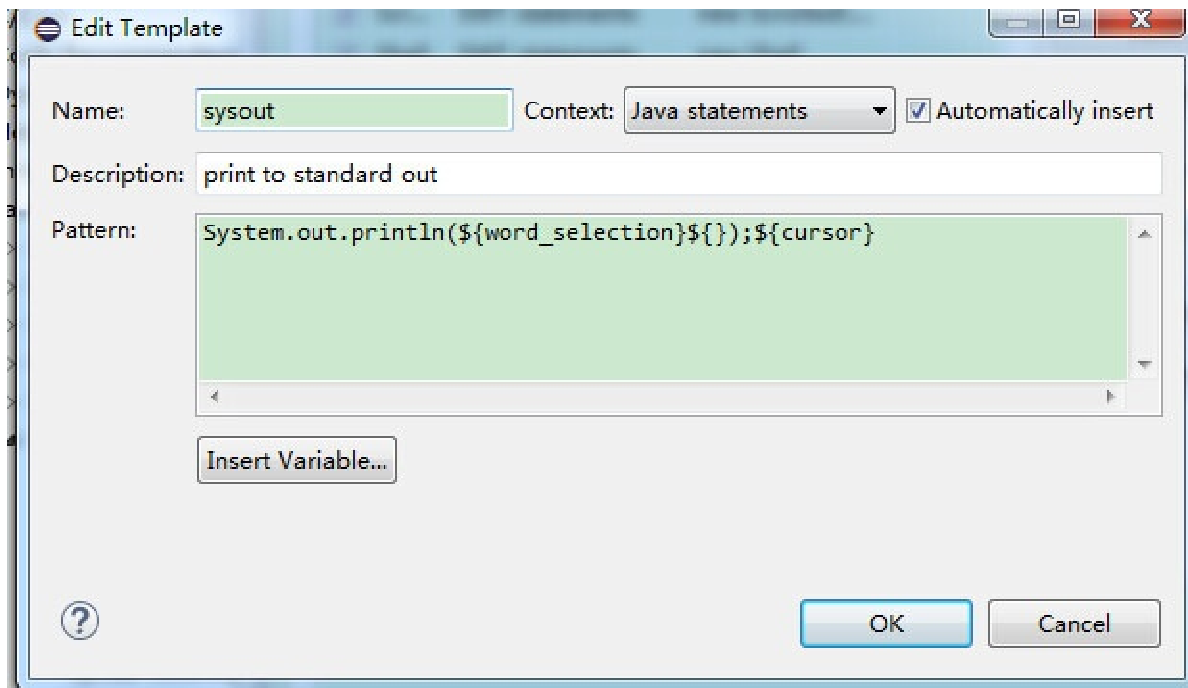


自定义代码模板

Eclipse 还提供了非常多的代码模板，我们可以通过 Windows->Preferences->Java->Editor->Templates (你可以在搜索框中输入Templates查找)看到所有已定义的代码模板列表。



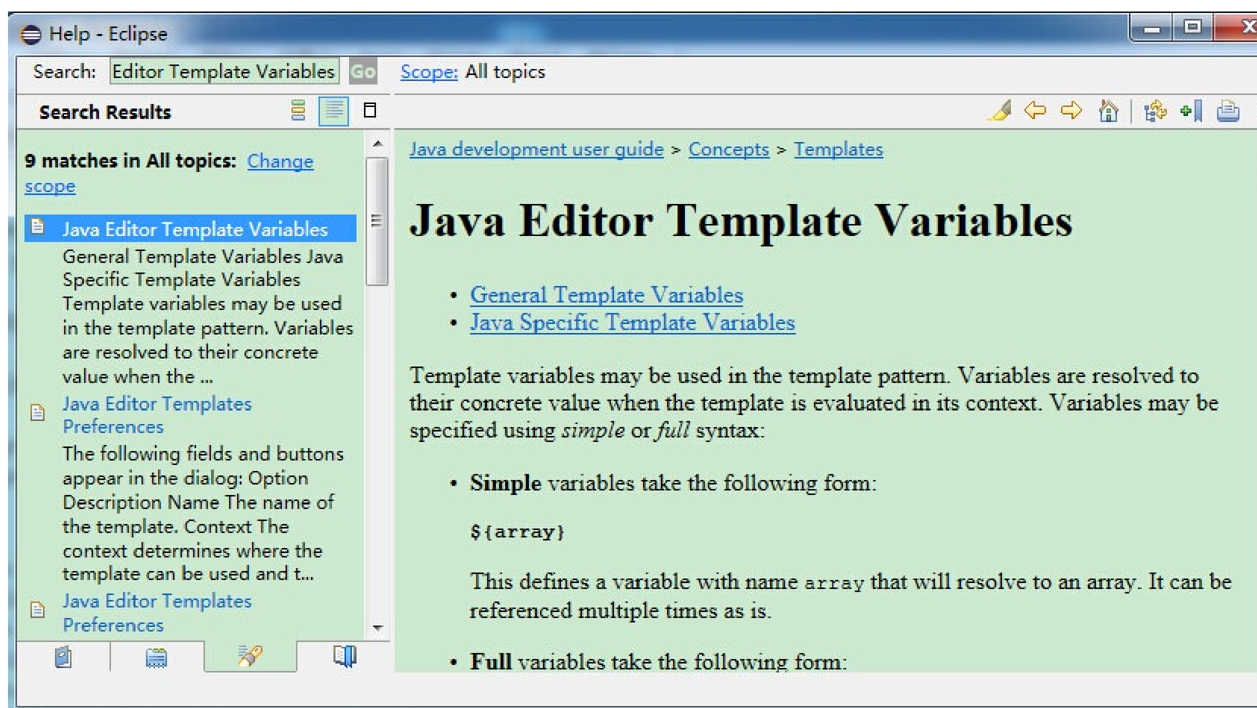
我们在弹窗口选中 sysout 模板并点击右侧Edit，显示如下：



编辑面板是核心关注对象，因为一切东西都在这里面配置。先来熟悉下这个面板中关键的五项分别是什么。

- **Name**：名称，其实就是以后可以用到的代码缩写
- **Context**：模板上下文，指定该代码模板在什么地方才能生效，对于Java至少包含这么四个：
  1. Java type members，模板对应的代码是类成员，psvm模板严格来说应该选择这个
  2. Java statements，模板对应的代码是语句块
  3. Java，最通用的，只要是Java代码就行
  4. Java doc，顾名思义了
- 模板变量：eclipse已经预置了一些模板变量（点Insert Varibles可以看到所有预置变量），如：
  1. ${cursor}是表示光标
  2. ${date}表示当前日期字符串
  3. ${time}表示当前时间字符串
  4. ${line_selection}让当前行被选中
  5. ${word_selection}让当前单词被选中当然我们也可以定义自己的模板变量，比如我定义一个 ${myTemplateVarible}，那么对应代码显示的就是 myTemplateVarible。
- **Pattern**：代码模板对应的模式，按照你希望代码的格式逐个输入即可

更多自定义代码模板的内容你可以通过点击 Help 菜单中的 Help Contents 选项，在弹出的对话框的搜索栏上输入 "Java Editor Template Variables" 选择 Java Editor Template Variables 查看具体的文档描述：

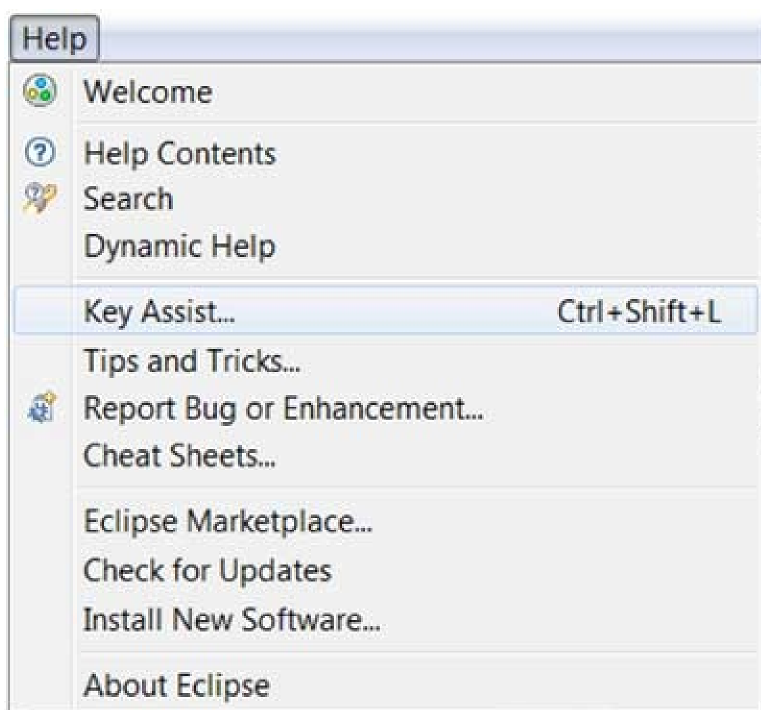

Help - Eclipse
Search: Editor Template Variables Go Scope: All topics
Search Results
9 matches in All topics: Change scope
Java Editor Template Variables
General Template Variables Java Specific Template Variables Template variables may be used in the template pattern. Variables are resolved to their concrete value when the ...
Java Editor Templates Preferences
The following fields and buttons appear in the dialog: Option Description Name The name of the template. Context The context determines where the template can be used and t...
Java Editor Templates Preferences
Java development user guide > Concepts > Templates
Java Editor Template Variables
General Template Variables
Java Specific Template Variables
Template variables may be used in the template pattern. Variables are resolved to their concrete value when the template is evaluated in its context. Variables may be specified using simple or full syntax:
Simple variables take the following form:
${array}
This defines a variable with name array that will resolve to an array. It can be referenced multiple times as is.
Full variables take the following form:

Eclipse 快捷键

关于快捷键

Eclipse 的很多操作都提供了快捷键功能，我们可以通过键盘就能很好的控制 Eclipse 各个功能：

- 使用快捷键关联菜单或菜单项
- 使用快捷键关联对话窗口或视图或编辑器
- 使用快捷键关联工具条上的功能按钮



Eclipse 快捷键列表可通过快捷键 Ctrl + Shift + L 打开 。

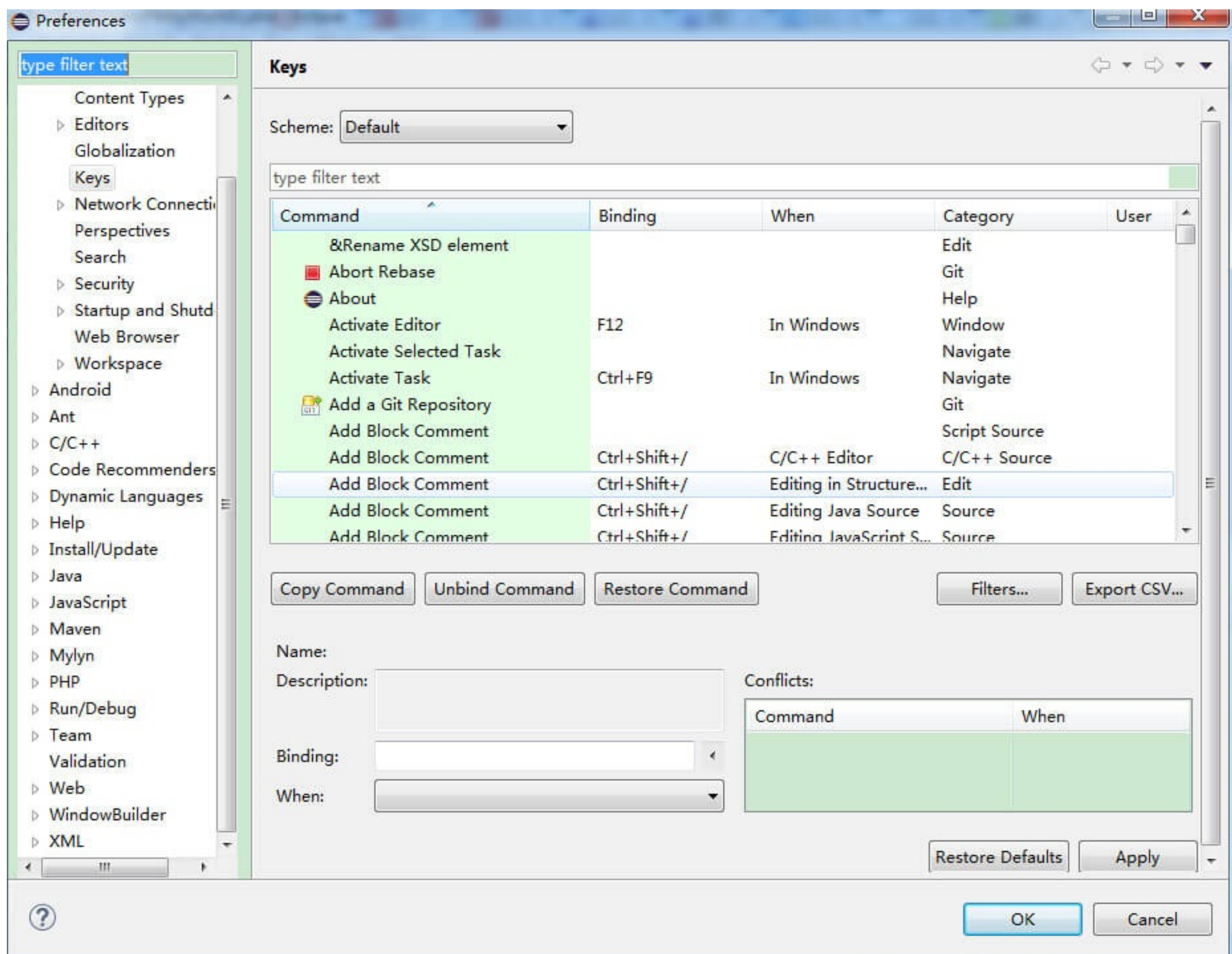
Activate Editor	F12
Activate Task	Ctrl+F9
Add Javadoc Comment	Alt+Shift+J
All Instances	Ctrl+Shift+N
Backward History	Alt+Left
Build All	Ctrl+B
Change Method Signature	Alt+Shift+C
Close	Ctrl+F4
Close All	Ctrl+Shift+F4
Collapse All	Ctrl+Shift+Numpad_Divide
Commit...	Ctrl+#
Content Assist	Ctrl+Space
Context Information	Ctrl+Shift+Space
Copy	Ctrl+Insert
Cut	Shift+Delete
Deactivate Task	Ctrl+Shift+F9
Debug	F11
Debug Ant Build	Alt+Shift+D, Q
Debug JUnit Test	Alt+Shift+D, T
Debug Java Applet	Alt+Shift+D, A
Debug Java Application	Alt+Shift+D, J

Press "Ctrl+Shift+L" to open the preference page.

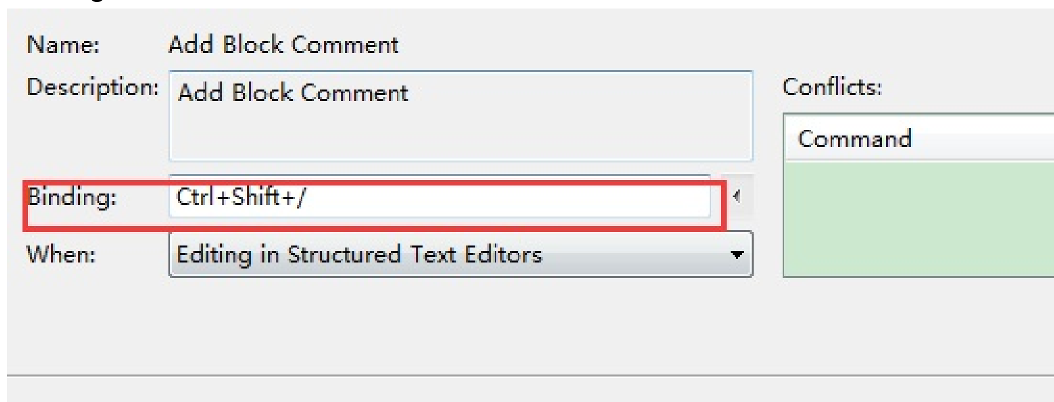
设置快捷键

Eclipse 系统提供的快捷键有时比较难记住，甚至根本没有提供快捷键时，就需要自己手动设置快捷键。

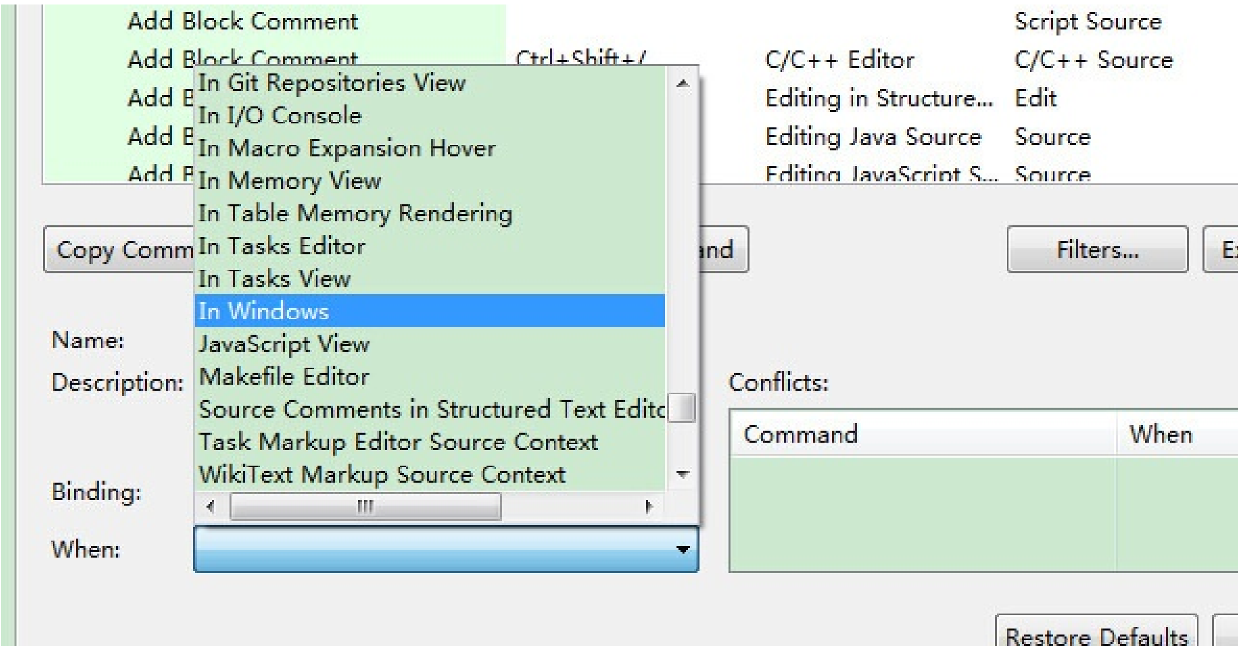
我们可以通过点击window->preferences->general->keys（或直接搜索keys），进入快捷键管理界面：



在这里可以查找所有功能的快捷键，需要修改或新增时，点击需要修改或新增的命令，在binding 里设置快捷键:



设置完快捷键后，还需要设置在什么时候可以使用该快捷键，eclipse提供各种场景供选择，一般选择In Windows(即在eclipse窗口激活状态)即可。



完成以上操作，点击 OK 按钮即完成设置。

Eclipse 常用快捷键

快捷键	描述
编辑	
Ctrl+1	快速修复（最经典的快捷键,就不用多说了，可以解决很多问题，比如import类、try catch包围等）
Ctrl+Shift+F	格式化当前代码
Ctrl+Shift+M	添加类的import导入
Ctrl+Shift+O	组织类的import导入（既有Ctrl+Shift+M的作用，又可以帮你去除没用的导入，很有用）
Ctrl+Y	重做（与撤销Ctrl+Z相反）
Alt+/	内容辅助（帮你省了多少次键盘敲打，太常用了）
Ctrl+D	删除当前行或者多行
Alt+↓	当前行和下面一行交互位置（特别实用,可以省去先剪切,再粘贴了）
Alt+↑	当前行和上面一行交互位置（同上）
Ctrl+Alt+↓	复制当前行到下一行（复制增加）
Ctrl+Alt+↑	复制当前行到上一行（复制增加）
Shift+Enter	在当前行的下一行插入空行（这时鼠标可以在当前行的任一位置,不一定是最后）
Ctrl+/	注释当前行,再按则取消注释

选择	
Alt+Shift+↑	选择封装元素
Alt+Shift+←	选择上一个元素
Alt+Shift+→	选择下一个元素
Shift+←	从光标处开始往左选择字符
Shift+→	从光标处开始往右选择字符
Ctrl+Shift+←	选中光标左边的单词
Ctrl+Shift+→	选中光标又边的单词
移动	
Ctrl+←	光标移到左边单词的开头，相当于vim的b
Ctrl+→	光标移到右边单词的末尾，相当于vim的e
搜索	
Ctrl+K	参照选中的Word快速定位到下一个（如果没有选中word，则搜索上一次使用搜索的word）
Ctrl+Shift+K	参照选中的Word快速定位到上一个
Ctrl+J	正向增量查找（按下Ctrl+J后,你所输入的每个字母编辑器都提供快速匹配定位到某个单词,如果没有,则在状态栏中显示没有找到了,查一个单词时,特别实用,要退出这个模式，按escape建）
Ctrl+Shift+J	反向增量查找（和上条相同,只不过是从后往前查）
Ctrl+Shift+U	列出所有包含字符串的行
Ctrl+H	打开搜索对话框
Ctrl+G	工作区中的声明
Ctrl+Shift+G	工作区中的引用
导航	
Ctrl+Shift+T	搜索类（包括工程和关联的第三jar包）
Ctrl+Shift+R	搜索工程中的文件
Ctrl+E	快速显示当前Editer的下拉列表（如果当前页面没有显示的用黑体表示）
F4	打开类型层次结构
F3	跳转到声明处
Alt+←	前一个编辑的页面
Alt+→	下一个编辑的页面（当然是针对上面那条来说了）

Ctrl+PageUp/PageDown	在编辑器中，切换已经打开的文件
调试	
F5	单步跳入
F6	单步跳过
F7	单步返回
F8	继续
Ctrl+Shift+D	显示变量的值
Ctrl+Shift+B	在当前行设置或者去掉断点
Ctrl+R	运行至行(超好用，可以节省好多的断点)
重构（一般重构的快捷键都是Alt+Shift开头的了）	
Alt+Shift+R	重命名方法名、属性或者变量名 （是我自己最爱用的一个了，尤其是变量和类的Rename,比手工方法能节省很多劳动力）
Alt+Shift+M	把一段函数内的代码抽取成方法 （这是重构里面最常用的方法之一了,尤其是对一大堆泥团代码有用）
Alt+Shift+C	修改函数结构（比较实用,有N个函数调用了这个方法,修改一次搞定）
Alt+Shift+L	抽取本地变量（ 可以直接把一些魔法数字和字符串抽取成一个变量,尤其是多处调用的时候）
Alt+Shift+F	把Class中的local变量变为field变量 （比较实用的功能）
Alt+Shift+I	合并变量（可能这样说有点不妥Inline）
Alt+Shift+V	移动函数和变量（不怎么常用）
Alt+Shift+Z	重构的后悔药（Undo）
其他	
Alt+Enter	显示当前选择资源的属性，windows下的查看文件的属性就是这个快捷键，通常用来查看文件在windows中的实际路径
Ctrl+↑	文本编辑器 上滚行
Ctrl+↓	文本编辑器 下滚行
Ctrl+M	最大化当前的Edit或View （再按则反之）
Ctrl+O	快速显示 OutLine（不开Outline窗口的同学，这个快捷键是必不可少的）
Ctrl+T	快速显示当前类的继承结构
Ctrl+W	关闭当前Editer（windows下关闭打开的对话框也是这个，还有qq、旺旺、浏览器等都是）

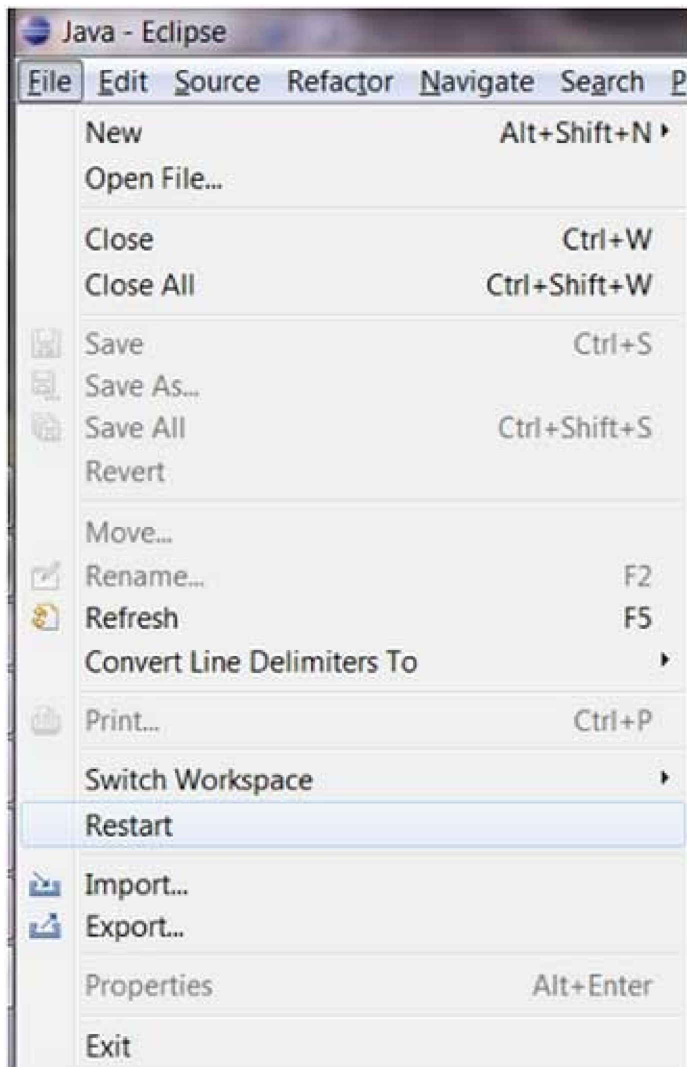
Ctrl+L	文本编辑器 转至行
F2	显示工具提示描述

Eclipse 重启选项

重启 Eclipse

重启选项允许用户重启 Eclipse。

我们可以通过点击 File 菜单选择 Restart 菜单项来重启 Eclipse。

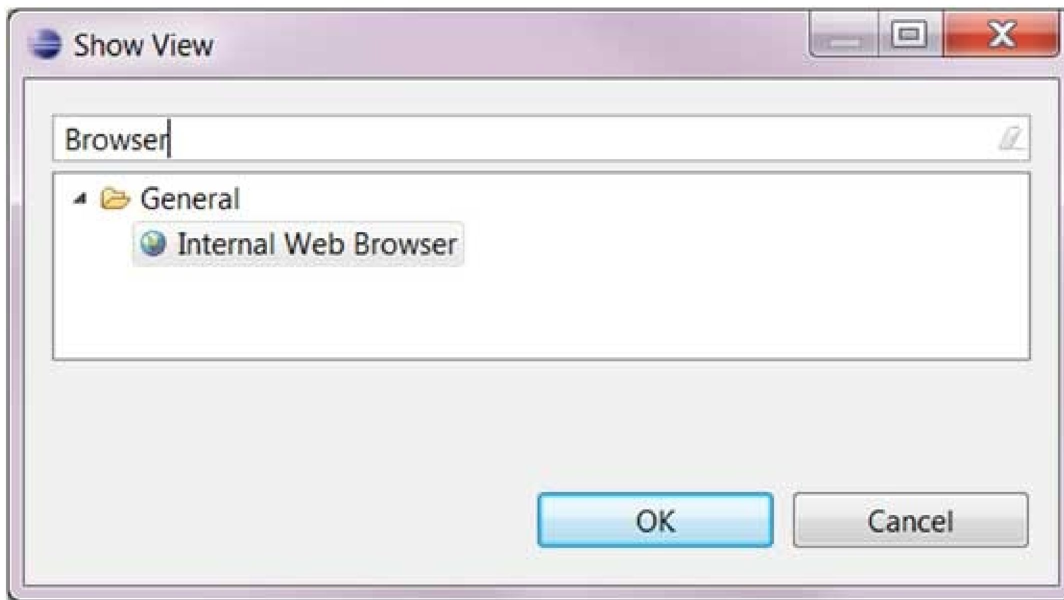


在安装插件后，用户一般都会被提醒要重启 Eclipse。如果用户当时没有重启Eclipse，可以通过该选项来重启。

Eclipse 内置浏览器

Web 浏览器

Eclipse 系统内部自带了浏览器，该浏览器可以通过点击 Window 菜单并选择 Show View > Other，在弹出来的对话框的搜索栏中输入 "browser"。



在树形菜单中选择 "Internal Web Browser" 并点击 OK。

在内置浏览器中我们在地址栏中输入网址，如：http://www.w3cschool.cc，即可打开网页。



免责声明

W3School提供的内容仅用于培训。我们不保证内容的正确性。通过使用本站内容随之而来的风险与本站无关。W3School简体中文版的所有内容仅供测试，对任何法律问题及风险不承担任何责任。